Des ressources aux traitements linguistiques : le rôle d'une architecture linguistique

Frederik Cailliau

Université Paris 13 – Paris Nord
Institut Galilée
Laboratoire d'Informatique de Paris-Nord – LIPN – UMR CNRS 7030

N° attribué par la bibliothèque
|__|__|__|__|__|__|__|__|__|__|

Des ressources aux traitements linguistiques : le rôle d'une architecture linguistique

Mise en place d'un environnement de gestion de ressources linguistiques pour une plate-forme d'analyse textuelle

THÈSE

pour obtenir le grade de

Docteur de l'Université Paris 13 – Paris Nord

discipline : informatique

présentée et soutenue publiquement le 9 décembre 2010 par

Frederik Cailliau

Membres du jury :

Rapporteurs :	M. Patrice Bellot	Université d'Avignon et des Pays de Vaucluse
	M. Nabil Hathout	CNRS, Université de Toulouse II – Le Mirail
Examinateurs:	M. Olivier Gaunet	Sinequa
	M. François Lévy	Université de Paris 13 – Paris-Nord
	M. Claude de Loupy	Syllabs
	M. Emmanuel Morin	Université de Nantes
Directeur :	Mme Adeline Nazarenko	Université de Paris 13 – Paris-Nord

PREFACE ET REMERCIEMENTS

De nombreuses années se sont écoulées depuis que je suis entré à Sinequa au début de 2002 au rythme de quelques jours par semaine. Après l'obtention du DEA à Paris 13, j'ai échangé mon tapis d'intermittent du TAL contre une chaise dans l'équipe de Sinequa Labs, grâce au contrat Cifre. Mon implication dans le développement des ressources des *nouvelles* langues me permet de gravir un échelon, et comme si cela ne me suffisait pas, on m'offre rapidement un deuxième bureau, devenu vacant, en tant que chef de projet des projets de recherche.

Cette nouvelle position me permet de voir un peu le pays, qu'il soit linguistique ou informatique ou un mélange juteux des deux. Les ressources vont bien, quoique le carcan imposé par l'architecture informatique soit étroit et tout reste à formaliser. Le passage brusque d'un système gérant quelques langues à de multiples langues se fait presque du jour au lendemain, alors que la terre n'a même pas été râclée et que son jardinier manque des outils les plus élémentaires.

Dans le désert de l'information, j'ai construit une tente qui est devenue une bibliothèque de documentation, et j'ai planté des graines d'outils qui sont devenues des baobabs vivant leur propre vie. Au final, l'endroit ne sera pas tout à fait devenu un oasis, mais il y fait bon vivre.

Dans mon environnement direct, les « *Mais pourquoi ça marche ?* » sont un peu moins nombreux que les « *Pourquoi ça (ne) marche pas ?* » pourtant exprimés avec le même brin d'indignation. Si Sinequa a beaucoup changé depuis que j'y suis, ces petites phrases montrent que nos échanges avec la machine en face n'ont pas tant changé. Elle est souvent réfractaire à nous obéir, surtout quand on n'arrive pas à parler sa langue proprement, ni à comprendre son retour. Mes travaux ont pour but de faciliter cette communication qui réunit utilisateurs et programmeurs afin d'améliorer le quotidien de chacun.

Non sans mal le capitaine à bord a su guider un navire qui préfère profiter du bon vent pour partir à la découverte du grand large plutôt que de se fier à la carte. Contrat Cifre oblige, les destinations ont été imposées par les besoins industriels de Sinequa, pas toujours compatibles avec les souhaits académiques ou les miens.

Le parcours d'un homme est celui de ses rencontres. Au fil de ces années, les rencontres ont été nombreuses, donnant lieu à d'innombrables histoires. Si ce n'est pas l'endroit de les raconter, il convient néanmoins de remercier tous les acteurs, quel que fût leur rôle. Je remercie chacun d'entre vous pour vos mots, vos gestes, votre présence, voire votre absence quand il le fallait.

Je tiens à exprimer tout d'abord ma gratitude envers les membres du jury : Nabil Hathout et Patrice Bellot, pour leurs remarques critiques et édifiantes en qualité de rapporteurs, Olivier Gaunet, François Lévy, Claude de Loupy et Emmanuel Morin, pour leur lecture attentive en tant qu'examinateurs, Adeline Nazarenko, pour avoir dirigé et encadré cette thèse dans les règles de l'art.

Aude, je te remercie pour avoir lu et relu et relu et relu.

Je remercie les membres et ex-membres de l'équipe de Sinequa pour leurs incessants encouragements, dans le désordre, avec beaucoup d'oublis : Philippe, Eric et Eric, feu Pascal, Kevin, Gaëlle, Luc, Olivier, Mélodie, Aurélia, les dirigeants actuels Jean et Alex, et tous les autres. Je remercie également toutes les personnes qui ont travaillé sur les ressources linguistiques à Sinequa, pour m'avoir fait entrevoir les rouages de leurs langues maternelles.

Je remercie les membres du LIPN pour leur accueil au cours de mes visites au labo.

Merci finalement à tous mes amis, ma famille et mon amie pour avoir supporté mes caprices de thésard : je ne promets pas qu'ils prendront fin, mais je vous le souhaite.

Pour la suite je laisse la parole au *nous d'auteur*. Quand il s'agit de travaux qui n'ont pas été faits par moi-même, j'abuserai d'autres moyens linguistiques pour l'indiquer, en utilisant des expressions comme *l'équipe linguistique* ou *l'équipe de Sinequa*.

Vous souhaitant une bonne lecture,

Frederik

TABLE DES MATIERES

INTRODUCTION

Depuis les débuts de l'intelligence artificielle, le rêve de *faire parler les machines* n'a jamais été très loin. Il ne s'agit pas simplement de parler, mais de maîtriser toute la communication, au point de confondre homme et machine. Face à la complexité de la langue, les enjeux sont devenus beaucoup plus réalistes. Des techniques de traitement automatique des langues (TAL) existent aujourd'hui un grand nombre d'applications : fouille de données, recherche d'information (RI), correction orthographique, aide à la traduction, gestion terminologique, etc.

Selon les méthodes mises en œuvre, les techniques de TAL reposent plus ou moins sur des ressources linguistiques, qu'on peut voir comme des sources d'information primaires, nécessaires au bon fonctionnement du système. En effet, pour savoir décoder le texte d'un document et le transformer en informations, voire en connaissances, il faut préalablement disposer de suffisamment d'informations linguistiques.

Le nombre d'informations ainsi nécessaires dépend des traitements et des méthodes mis en œuvre. Sans pour autant procéder à des traitements très évolués, le nombre d'informations peut rapidement devenir impressionnant, et d'autant plus s'il y a utilisation de lexiques. Les lexiques morphosyntaxiques, par exemple, référencent les mots de la langue et leur attachent des informations grammaticales comme la catégorie grammaticale, le genre, le nombre, etc. Des lexiques de couverture générale de la langue commune peuvent avoir plusieurs centaines de milliers d'entrées. Selon les caractéristiques de la langue, ce chiffre varie énormément, d'autant qu'un certain nombre d'informations peuvent être codées sous forme de règles lexicales. Les traitements qui dépassent le niveau lexical peuvent également reposer sur des ressources, souvent sous forme de grammaires.

Les ressources linguistiques sont spécifiques à la langue et diffèrent donc d'une langue à l'autre. Leur élaboration demande un investissement qui peut être considérable, et augmente linéairement avec le nombre de langues. Plus les traitements sont évolués, plus les ressources risquent d'être nombreuses et complexes. Le nombre élevé d'informations contenues dans les ressources complique également la mise à jour. Ces problèmes de gestion demandent des solutions pour faciliter le codage et garantir la cohérence entre les différentes ressources linguistiques d'une même langue.

Les outils que nous avons mis en place constituent un environnement de gestion qui répond à la problématique de gestion d'une grande masse d'informations linguistiques. Dans notre cas, les traitements qui exploitent ces informations augmentent les performances d'un moteur de recherche industriel. Ils procèdent à une analyse textuelle avancée, et disposent de ressources linguistiques pour 19 langues.

Le module d'analyse textuelle qui est au centre du moteur de recherche de Sinequa est une plate-forme dont les traitements ont été mis en place progressivement. Chaque traitement ayant ses propres besoins en informations linguistiques, son ajout dans la chaîne entraîne des modifications dans les ressources linguistiques. Les ressources linguistiques ont ainsi été constituées et enrichies au fur et à mesure que les besoins d'analyse sont devenus plus importants. Les premiers lexiques de Sinequa ont été mis en place il y a plus de 20 ans, et le

module d'analyse n'a cessé d'évoluer depuis, suivant l'innovation du produit. De nombreuses informations se sont ajoutées depuis, et des ressources supplémentaires, avec parfois des informations redondantes, ont vu le jour.

Pour la gestion des ressources, une vue d'ensemble est indispensable. Une compréhension profonde du fonctionnement du système est nécessaire pour garantir la cohésion entre les traitements, les ressources et les outils de gestion. Elle s'obtient par la formalisation de ce que nous appelons l'*architecture linguistique* du système. Cette description formelle des entrées et des sorties des traitements, des informations prises dans les ressources influant sur les traitements et des connaissances produites permet d'obtenir la vision globale du système. Cette formalisation permet de répondre à la question à savoir quels traitements sont effectués à quel moment avec quelles informations provenant de quelles ressources.

Au-delà du développement d'outils de gestion, la compréhension de l'architecture linguistique permet en outre d'imaginer des traitements supplémentaires ou de mieux situer les problèmes quand ils se présentent. Le système sur lequel nous avons travaillé a évolué organiquement, avec l'architecte du système en gardien de la cohésion entre ressources et traitements. Néanmoins, les plans de l'architecture n'ayant jamais été explicités, il a toujours existé un écart entre la compréhension des linguistes, fournisseurs de ressources, et celle des informaticiens, architectes des traitements. Il est vrai que leurs visions sur les ressources sont profondément différentes. Les linguistes voient généralement dans les ressources linguistiques une représentation partielle de la langue. L'imperfection étant inhérente aux ressources, elles ne peuvent être stabilisées, l'essence d'une langue ne pouvant pas être captée en quelques fichiers. Pour les informaticiens, les ressources linguistiques sont souvent de simples sources d'informations qui sont exploitées par les traitements qu'ils ont conçus, la gestion de ces ressources étant secondaire, et leur modification à éviter.

Pourtant, le respect de l'architecture linguistique est d'autant plus important qu'il existe des contraintes industrielles sur le système, ajoutant un historique pesant et des formats quasiment immuables. Grâce à la vision que procure la formalisation de l'architecture linguistique il est possible de mettre en place des suites de tests de non régression précises pour chaque traitement et de garantir ainsi la qualité des traitements et du système de manière globale. L'ensemble des outils qui constituent l'environnement de gestion doivent faciliter toutes les étapes de gestion des ressources linguistiques. Leur but est de rendre plus simple les tâches d'acquisition et de mise à jour, qui sont par définition complexes, à cause de la masse de données et la diversité des sources. Ce faisant, nous aspirons à augmenter la qualité des ressources par une réduction du nombre d'erreurs et une cohérence accrue entre les informations qu'elles contiennent.

Notre thèse est organisée en neuf chapitres, assemblés en quatre parties.

La **première partie**, « **Le document, le TAL et la RI** » contient deux chapitres. Le premier chapitre est une brève introduction sur les documents, comme support de la communication, et les signes, comme matière première de cette communication. Le chapitre 2 situe le cadre de l'exploitation des ressources, qui est la recherche d'information dans notre cas. Il introduit l'architecture du moteur de recherche et illustre comment des techniques de TAL peuvent améliorer la recherche d'information. Finalement, nous faisons état du grand nombre de ressources que ces techniques impliquent, ainsi que des contraintes industrielles qui imposent des conditions de travail particulières.

La **deuxième partie**, « **Etat de l'art et problématique** » consiste en deux chapitres qui couvrent un état de l'art sur deux sujets liés. Le chapitre 3 est consacré aux ressources linguistiques, leur cycle de vie et les outils qui permettent de les gérer. Le chapitre 4 retrace les travaux sur la formalisation lexicale, qui a une longue tradition et atteint son point d'orgue avec la norme ISO récemment approuvée.

La **troisième partie**, « **Modélisation d'une architecture linguistique** », présente en deux chapitres notre modélisation des techniques de traitement automatique des langues utilisées dans le moteur. A partir des différentes approches en vigueur, nous formalisons dans le chapitre 5 de façon générale les traitements de TAL éprouvés dans le cadre de la RI, ainsi que les connaissances qui leur sont indispensables pour fournir le résultat voulu. Dans le chapitre 6, nous présentons comment ce modèle s'applique aux traitements et connaissances sur le cas concret du module d'analyse textuelle du moteur de recherche de Sinequa par la formalisation de son architecture linguistique.

La **quatrième partie**, « **Création d'un environnement de gestion de ressources** », présente la mise en place d'un environnement de gestion de ressources en trois chapitres. Le chapitre 7 présente les différents outils que nous avons mis en place pour améliorer la gestion des ressources. L'accent est mis sur le prototype qui résulte de la modélisation et qui offre un accès unique et global aux ressources lexicales. Le chapitre 8 présente les outils de pilotage de l'environnement. Ils servent à mieux comprendre l'état des ressources et leur évolution pour mieux cerner les besoins et planifier les futurs travaux. Le chapitre 9 traite de l'acquisition de ressources lexicales et de leur formatage en intension ou en extension. Il présente ensuite une aide expérimentale à l'enrichissement lexical qui se base sur les informations disponibles dans les lexiques extensionnels de Sinequa.

Nous terminons la thèse avec la conclusion et les perspectives, suivies par la bibliographie et les annexes.

PARTIE 1

Le document, le TAL et la RI

Chapitre 1

Des signes aux connaissances

La communication est une partie essentielle de la vie. L'être humain communique par des signes qu'il fige parfois sur des documents qui mènent ensuite leur propre vie. La langue joue un rôle primordial dans la communication, *a fortiori* quand elle est textuelle. Le traitement automatique de ce système complexe se heurte à de nombreux obstacles, dont le tout premier est le choix de l'unité de traitement. Pour mieux comprendre la difficulté et les conséquences de ce choix, nous explorons dans ce chapitre l'articulation entre le document, les signes textuels qu'il peut comporter et l'unité de traitement.

1.1 Des signes pour communiquer

Comme tout système de communication, la langue manipule des signes. Ce sont des unités qui ont leur propre sémantique, et qui subissent la linéarité du temps. Leur signification leur est donnée par la communauté qui les utilise et peut changer selon le contexte, c'est-à-dire l'endroit et la situation dans lesquels ils sont employés.

Le traitement de la communication par des procédés informatiques se heurte principalement à l'ambiguïté des unités qui sont manipulées et la difficulté de construire le sens qui n'est pas simplement compositionnel. Comme pour tout langage, la signification de signes va être construite par la combinaison des mêmes facteurs : le sens qui lui est donné par la communauté de locuteurs (sémantique lexicale), la syntaxe (ordre des signes) et la situation de communication dans lesquelles ils sont utilisés (dimension pragmatique).

Pour construire une machine intelligente conversationnelle qui passerait le test de Turing, il faudrait savoir interpréter tout type de signe, savoir traiter les informations d'ordre pragmatique, savoir les représenter et raisonner dessus. L'état de l'art des différents sous-domaines contributoires à l'Intelligence Artificielle (IA) en est encore loin, même si de grands progrès ont été faits.

Les signes que nous avons pu traiter au cours de notre thèse sont des signes verbaux, car nos travaux se situent dans le domaine du Traitement Automatique des Langues (TAL) et aux abords de la Recherche d'Information (RI). Cela couvre ce qu'on appelle de l'écrit : des informations textuelles, qu'elles obéissent ou non aux règles syntaxiques de la langue en question. Mais cela couvre aussi des transcriptions de productions orales : principalement du texte lu (bulletin d'informations) et des conversations téléphoniques de centres d'appel.

1.2 Le document et les signes

En science de l'information et de la documentation, le document est défini comme la combinaison d'un support et d'une information, les deux étant inextricablement liés : l'information exige la présence d'un support, et tout support est susceptible de porter des informations. La réflexion lancée par le réseau thématique pluridisciplinaire RTP-DOC propose de distinguer le document comme forme, comme signe et comme médium. En TAL et en IA en général, le document est en effet tout d'abord considéré comme « un objet signifiant, et est intéressant pour le traitement du contenu. Si la forme est parfois prise en compte, elle ne l'est que comme porteuse de sens » ([Pédauque, 2006]).

La lecture des documents par un système informatique se fait linéairement, ce qui crée un flux informationnel. Le traitement automatique de documents demande un découpage bas niveau de ce flux en signes. Ces signes donnent accès à des connaissances après interprétation. Le texte peut alors être représenté par des connaissances exploitables par le système. [Pédauque, 2006] le formule ainsi : « A leur manière [informaticiens et linguistes] ont suivi un chemin parallèle à celui des documentalistes, reconstruisant à l'aide de filtres et de calculs, sinon un langage au sens informatique, du moins du texte censé représenter d'une façon structurée le contenu d'un document. »

Plusieurs candidats se présentent comme possibles unités de traitement, des suites de signes aux composants des signes. Pour les documents textuels, ils correspondent aux niveaux de structuration des documents et des signes : paragraphes, suites de phrases, phrases, suites de mots, mots, composants de mots, morphèmes et caractères. Tout choix a de fortes

implications sur le traitement, la gestion des connaissances, et peut limiter les applications possibles.

Cela nous mène aux questions suivantes :

- Quelle est l'unité de traitement idéale ?
- Comment organiser les connaissances de façon optimale ?

Les signes d'un document peuvent être de diverses natures : image, son, vidéo ou texte, voire une combinaison de ceux-ci. Même si on peut argumenter que l'unité de traitement idéale dépend du type de traitement qu'on envisage, ce n'est pas un hasard si en TAL, le *mot* a été choisi comme interface entre les documents et les connaissances, car c'est l'unité qui correspond au signe dans la plupart des cas. Un grand nombre de traitements, mobilisant parfois des ressources, est mis en œuvre pour arriver à identifier ces unités et découper en mots le flux informationnel, qui – faut-il le rappeler ? – ne correspond pour l'ordinateur qu'à une série de chiffres.

Les mots donnent accès aux informations qui leur sont associées dans des listes qu'on appelle des *dictionnaires* ou des *lexique*s. Ces listes servent de ressources au traitement qui effectue l'analyse des documents. L'ensemble des ressources peut être désigné comme une *base de connaissances*.

Dans cette thèse, nous nous intéressons en premier lieu aux documents numériques dont l'information est faite de signes textuels. Parmi ces signes, ceux qui sont porteurs de sens ou modifieurs de ces derniers (= généralement appelés les *mots grammaticaux*) sont des *signes linguistiques*.

Dans la plupart des systèmes d'écriture, deux autres types de signes textuels séparent les signes linguistiques entre eux : il s'agit des ponctuations et des signes d'espacement qu'on appelle aussi *séparateurs*. La ponctuation est plus ou moins standardisée dans les systèmes d'écriture des langues occidentales, toutes alphabétiques. Pour cette raison on arrive facilement à lire la structure d'un document dans ces langues. Même si certains signes de ponctuation peuvent modifier la modalité d'un ensemble de mots, par exemple par l'ajout d'un point d'interrogation, nous ne les considérons pas comme des signes linguistiques. La ponctuation est régie par les règles d'orthographe, plus ou moins strictes selon la langue, et non pas par la grammaire. Elle n'existe que dans la langue écrite et sert à donner une structure au document pour augmenter la lisibilité.

Dans l'évolution des systèmes d'écriture, on constate une évolution naturelle de signes dessinés vers des symboles, qu'on peut suivre sur la chronologie de [Fayet-Scribe, 1997]. Les dessins deviennent pictogrammes, qui rapidement se schématisent en idéogrammes et plus tard en caractères syllabiques et alphabétiques. Avec l'introduction des caractères, on donne une structure linéaire interne aux signes en faisant correspondre les caractères aux sons prononcés.

L'espace est un séparateur naturel et économique pour délimiter les signes, sans lequel la lisibilité fléchit. Ce séparateur a disparu temporairement avec l'adoption de la *scriptura continua* par les grecs et ensuite les romains. D'après [Llamas Pombo, 2001] ce sont les moines irlandais qui, entre le 6^{e} et 7^{e} siècle, commencent à introduire l'espace blanc comme une aide à la lecture dans les textes latins, habitude qui est reprise sur le continent à partir du 10^{e} siècle. La charge intellectuelle de découpage en mots revient au copiste et le lecteur en est libéré, ce qui favorise la lecture silencieuse et augmente la rapidité de lecture.

Les signes linguistiques sont communément appelés *mots*, terme imprécis pour lequel il manque une définition formelle. Une définition à partir de la forme est impossible car signes

d'espacement et de ponctuation sont ambigus, comme l'illustrent les simples exemples français suivants : *essuie-glace*, *pomme de terre*, *aujourd'hui*, *planche à voile*. Nul ne contestera, avec peut-être une légère hésitation qui se dissipe rapidement pour le dernier, que chacun de ces mots est un *mot* à part entière et que tous sauf le troisième sont constitués d'autres *mots*. Pour cette raison, il est impossible de s'appuyer sur les caractères et les règles orthographiques pour formaliser ce qu'est un *mot*.

Dans certains systèmes d'écriture, le découpage n'est pas formalisé par des séparateurs. L'exemple du thaï[1], qui utilise pourtant des caractères alphabétiques, montre que toutes les écritures n'ont pas de séparateur entre les signes. Ce n'est pas le cas non plus en chinois, mais pour une raison différente : le système d'écriture, fondé sur la juxtaposition d'idéogrammes, n'est pas alphabétique.

TH หาหน้าเว็บที่คล้ายกับหน้านี้

หา chercher หน้า page เว็บ web ที่ qui คล้าย ressemble กับ à หน้า page นี้ cette = rechercher les pages similaires à cette page

ZH 搜索类似以下网页的网页

搜索 chercher 类似 similaire 以下 suivant 网页 page web 的 de 网页 page web = rechercher les pages similaires à cette page

1.3 Le signe comme unité de traitement

Quel que soit le type de système d'écriture, l'analyse automatique des textes doit découper le flux de données jusqu'à manipuler l'unité minimale qu'elle retrouvera dans sa base de connaissances avant de pouvoir analyser le texte.

Le signe linguistique est généralement choisi comme unité minimale pour les traitements automatiques de la langue. Il dispose d'une autonomie qui est vérifiable par des tests linguistiques comme le test de la commutation utilisant les critères de l'analyse distributionnelle[2]. Un locuteur natif de n'importe quelle langue découpera intuitivement et sans hésiter une production orale en signes linguistiques, qu'il désignera comme *mots*. Il remplacera intuitivement un mot par un autre, sans faire de distinction entre mot, mot composé, mot dérivé ou autre. Il recourra au test de commutation s'il faut motiver formellement les choix qu'il a faits et se basera sur son expérience et sa connaissance de la langue. Dans la langue écrite ce découpage naturel en signes linguistiques se formalise dans la plupart des systèmes d'écriture par l'utilisation de l'espace et de signes de ponctuation entre les signes.

Il semble alors tout à fait naturel de choisir le signe linguistique comme unité du lexique qui fait le lien entre le texte analysé et les connaissances. Grâce aux séparateurs, il est facilement identifiable dans le flux. Il est modulaire dans le sens où on peut facilement recomposer les

[1] Nous nous servons des abréviations de la norme ISO 639-1 pour indiquer la langue des exemples. La liste des abréviations utilisées peut être trouvée en Annexe K, p. 289.

[2] Ce test s'effectue en remplaçant une unité par une autre en vérifiant que la phrase est toujours syntaxiquement et sémantiquement cohérente.

mots-formes complexes du lexique – s'il y en a – en recollant les suites de signes linguistiques. En absence de séparateurs, une étape de traitement supplémentaire s'impose pour décomposer le flux en signes. C'est le cas en thaï comme on l'a vu dans l'exemple ci-dessus : un signe textuel y égale une phrase qu'on découpe en une suite de signes linguistiques pour faire le lien avec les connaissances.

Le cas est un peu différent en chinois où chaque idéogramme égale un signe. L'exemple suivant est pris de [Emerson, 2000] et illustre la façon de construire le sens en chinois. Chaque idéogramme peut être interprété séparément pour la construction du sens global de la phrase, mais un locuteur chinois privilégiera un découpage de la phrase en regroupant plusieurs signes.

我不是中国人

我 je 不 non 是 être 中 milieu 国 terre 人 personne.

Les regroupements possibles – et concurrents – pour les trois derniers idéogrammes sont les suivants, mais seul le dernier est sémantiquement valable.

中 milieu 国人 compatriote

中国 Chine 人 personne

中国人 Chinois (personne de nationalité chinoise)

Un signe peut en cacher d'autres, ce qui est le cas des amalgames. Ainsi, en français, *du* égale aux signes *de* plus *le*. Même phénomène en polonais avec les contractions de préposition et du pronom masculin singulier lui : *do niego* en *doń* (à lui), *przez niego* en *przezeń* (par lui). C'est aussi le cas des abréviations, comme *TGV* pour le cocktail *Téquila Gin Vodka*, ainsi que des élisions comme *l'* en français pour *le* ou *la*, ou *'t* en néerlandais pour *het* (article neutre).

Du point de vue de la sémantique, le signe linguistique n'est pas un bon candidat en tant qu'unité de traitement car il est fortement ambigu. Il suffit de penser à l'exemple emblématique d'*avocat*, fruit ou personne, pour s'en rendre compte.

1.4 Le contenu de la base de connaissances

La base de connaissances contient d'une part les informations nécessaires pour distinguer les signes linguistiques dans le flux textuel, et d'autre part des informations sur ces signes. Ces informations seront réexploitées et deviendront alors des connaissances. La base contient des lexiques avec notamment des *mots-formes* qui servent à découper le flux en *unités textuelles* et à établir le lien avec les informations. Elle contient également des grammaires avec des règles générales d'analyse textuelle.

La forme des unités textuelles et des mots-formes est identique. Leur rôle n'est néanmoins pas du tout le même : les premiers contribuent à la construction d'un message, alors que les seconds ne sont que des pointeurs vers les informations de la base.

Le découpage du flux textuel n'est pas évident et peut aller au-delà de la simple comparaison d'une chaîne de caractères entre signes et mots-formes de la base de connaissances. Les mots-formes du lexique peuvent être complexes (contenir des espaces, des tirets, voire des ponctuations), auquel cas il faut prendre en compte plusieurs signes successifs, comme dans les exemples suivants :

FR	*base de données* *base de données relationnelle* *système de gestion de base de données* *casse-tête*	
ZH	网页	(page web)
DE	*je nachdem, ob*	(selon le cas où)
EL	ό,τι	(quiconque)
FI	EU:n	(de l'UE)

Dans certains cas, plusieurs découpages du texte sont possibles, comme l'illustrent les deux exemples suivants assez classiques :

a *pomme de terre cuite*
b *pomme de pin parasol*

Dans l'exemple a, l'ambiguïté de découpage est syntaxique et en même temps sémantique : faut-il l'interpréter comme une « pomme de terre qui est cuite » ou comme une « pomme en terre cuite » ?

Dans l'exemple b, l'ambiguïté est seulement syntaxique : faut-il le découper en *pomme de pin + parasol* ou en *pomme + pin parasol* ? Les performances requises pour certaines applications ne permettent pas de garder l'ambiguïté du découpage. L'un des deux découpages sera donc imposé par le traitement et les ressources qu'il exploite.

La base peut contenir en plus des informations sur la structure interne des signes s'ils sont décomposables en d'autres éléments. Elles peuvent être explicites (indiquant les composants) ou implicites (à base de règles). Différents types de connaissances morphologiques peuvent être distingués.

Le premier type concerne la *dérivation* ou *affixation dérivationnelle* : le mot dérivé se décompose en un élément non autonome et un élément autonome. Ce niveau couvre aussi bien la dérivation suffixale que préfixale. Les désinences sont souvent comptées dans les suffixes à cause de leur position, mais il faut bien distinguer la flexion de la dérivation.

FR	*prévisualisation* pré | visualisation	 (dérivation préfixale)
FR	*tirage* tirer | age	 (dérivation suffixale)
DE	*Lehrer* lehren | er	(nom : professeur) (apprendre| suffixe nominal)
RU	двадцатибазовый двадцати | баз | овый	(adjectif : en base douze) (douze | base | suffixe adjectival)
	десятимиллиардный десяти | миллиард | ный	(adjectif : dix milliardième) (dix | milliard | suffixe d'adjectif ordinal)

Le deuxième type concerne la *flexion* ou *affixation flexionnelle* : la forme fléchie se décompose en *base* plus *désinence*.

FR	*mangea*	mange | a	
NL	*Marks*	Mark | s	(de Mark)
FI	*ikkunallani*	ikkunalla | ni	(à la fenêtre)
FI	*EU:n*	EU | n	(de l'UE)

Le troisième type concerne la *composition* : le mot composé se décompose en plusieurs éléments autonomes.

EN	*database*	data | base	(base de données)
DE	*Energieeffizienz*	Energie | Effizienz	(efficacité énergétique)
	Zweifamilienhaus	zwei | Familie | Haus	(maison pour deux familles)
NL	*tweegezinswoning*	twee | gezin | woning	(maison pour deux familles)
FI	*kaupunkisuunnitteluvirasto* kaupunki | suunnittelu | virasto		(département de planification urbaine)

Le quatrième type concerne l'*insertion d'endoclitiques*, et l'*agglutination*. Dans ce dernier cas le mot contient un ou plusieurs clitiques qui s'affixent à des mots autonomes. L'exemple finnois montre l'affixation du clitique possessif *ni* à un nom au cas adessif. L'exemple en espagnol montre l'insertion des pronoms *te* et *lo* est pris de [Nevis, 2000], p. 390.

FI	*ikkunallani* ikkunalla | ni	(à la fenêtre | moi = à ma fenêtre)
ES	*[Quiero] mandar-te-lo* envoyer | à toi | le	([Je veux] te l'envoyer)

L'utilisation de ces connaissances pour la décomposition est illustrée dans la figure 1.

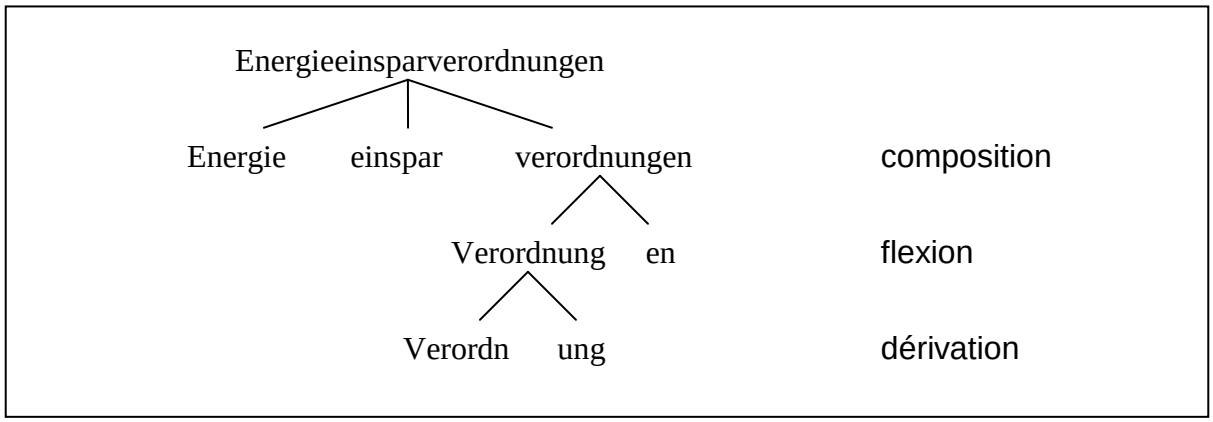


Figure 1: arbre morphologique du mot Energieeinsparverordnungen

Selon les besoins des traitements du système, le niveau de description contenu dans les ressources sera différent. Il en sera de même pour les informations associées. On peut ainsi associer des informations morphologiques, syntaxiques, ou sémantiques qui sont plus ou moins détaillées. Les informations obtenues à l'analyse peuvent être réutilisées dans des grammaires évoluées et servir à des analyses plus profondes.

1.5 Conclusion

Dans ce chapitre nous avons montré que le choix de l'unité de traitement pour le traitement automatique des langues est très important sans être évident. Un système multilingue doit en outre tenir compte des caractéristiques des langues qu'il traite, et cela à tous les niveaux d'analyse. La prise en compte de ces particularités nécessite une généricité dans la structuration des traitements et des ressources linguistiques. Le choix de l'unité conditionne l'architecture des traitements et des ressources linguistiques, actuels et futurs. Les systèmes complexes utilisent des unités de traitement différentes selon les traitements. Dans ces cas, la cohérence entre les informations des différentes ressources linguistiques doit être garantie. Face à la masse de données, ce n'est qu'avec les outils de gestion appropriés que la cohérence entre les ressources puisse être préservée.

Chapitre 2

Analyser pour retrouver : le TAL au service de la RI

La plateforme d'analyse linguistique que nous avons étudiée fait partie d'un moteur de recherche industriel. En réponse aux défis linguistiques que pose la recherche d'information, les architectes du moteur de recherche ont intégré des traitements en provenance du traitement automatique des langues (TAL). L'utilisation de techniques de TAL ne fait pourtant pas l'unanimité dans la communauté scientifique de la recherche d'information (RI). Dans ce chapitre nous présentons la place que prend l'analyse linguistique dans le moteur de recherche. Nous illustrons pourquoi ces traitements linguistiques ont une telle importance aux yeux des architectes du système, alors qu'ils mobilisent un nombre très important de ressources linguistiques qu'il faut gérer en tenant compte des contraintes économiques et industrielles.

2.1 La RI et le TAL, un mariage de raison

2.1.1 Des critiques et de leurs détracteurs

La Recherche d'Information (RI) est le domaine d'étude qui a comme objet d'étude la recherche dans des données non structurées. Pour des raisons historiques, les recherches ont tout d'abord porté sur les données textuelles. Le développement d'Internet aidant, les recherches dans les images, la vidéo et l'audio sont devenues autant de champs d'étude avec leurs spécialistes respectifs. Si nos recherches se limitent aux données textuelles, celles-ci peuvent également être des transcriptions d'autres supports, notamment d'audio et de video.

Le paradigme de la recherche d'information est bâti sur l'appariement entre une représentation de la requête de l'utilisateur et celle des documents, comme nous l'illustrons dans la figure 2 inspirée d'un schéma de [Rijsbergen, 2006]. Les systèmes qui sont mis au point intègrent un modèle de RI, qui explique la façon particulière de calculer la pertinence d'un document par rapport à une requête posée. Il existe une panoplie de modèles de RI qui ont été élaborés et testés au cours du temps : booléen, vectoriel, probabilistes, logiques, etc. L'une des difficultés clés en RI est la compréhension de l'information transmise et sa représentation.

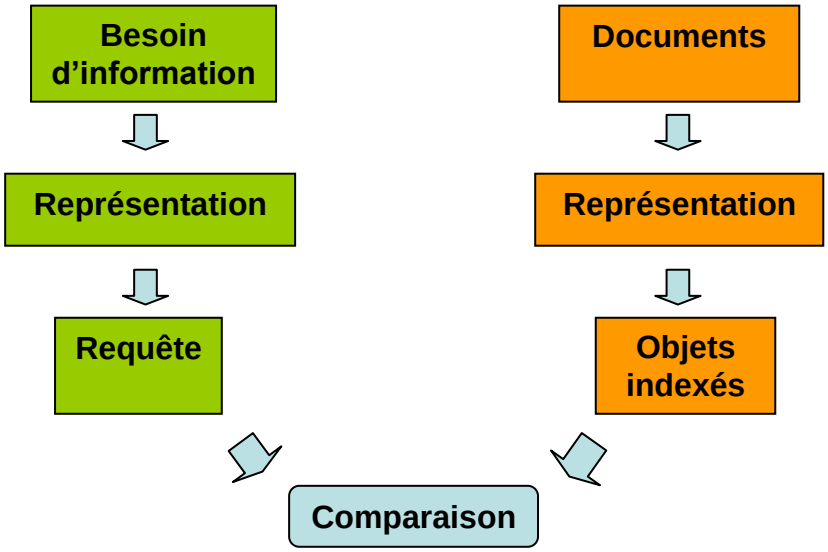


Figure 2 : La recherche d'information [Rijsbergen, 2006]

La langue étant un des principaux vecteurs de communication, il semblerait logique qu'une meilleure compréhension des sources textuelles passe obligatoirement par une compréhension profonde des mécanismes linguistiques qui servent de vecteur de communication. Or les expériences semblent démontrer qu'une analyse purement statistique suffit pour répondre aux besoins de la RI. [Harman, 1991] est l'un des premiers détracteurs qui, en comparant trois types de stemming dans un même système de RI, conclut que leur utilisation fait autant de bien que de mal : pour les recherches qui bénéficient du traitement, il y en a autant qui en souffrent.

Face aux critiques, nous faisons plusieurs constats. Le premier est que la plupart des expériences qui rejettent l'inclusion de techniques de TAL dans le modèle de RI ont été exécutées en anglais. Or les approches qui sont couronnées de succès en anglais ne le sont pas forcément dans d'autres langues à cause des caractéristiques linguistiques et orthographiques,

qui varient selon les langues. La variation morphologique, syntaxique et sémantique parmi les langues est telle que l'application d'un même modèle statistique ne peut qu'échouer si elle n'intègre pas une normalisation linguistique dans ses calculs. Le test de la version arabe de Google dans [Abbès et Boualem, 2008] est très parlant à ce sujet : un traitement de surface qui ignore les particularités linguistiques de l'arabe est voué à l'échec. L'analyse morphologique est insuffisante voire erronée. Il en est de même pour des langues bien moins « exotiques ». Ainsi, en allemand, la variation morphologique des cas et la composition pour la formation des mots sont de vrais obstacles pour une RI uniquement statistique. Dans le cadre du projet MuchMore[3], [Volk et al., 2002] montrent qu'une lemmatisation combinée avec une décomposition élémentaire des mots apporte une nette amélioration dans la recherche d'information en allemand. Par rapport à une recherche sans ces traitements, le rappel augmente de 60%, et la précision moyenne passe de 0,16 à 0,22.

Le deuxième constat est que malgré les critiques, le champ d'étude est toujours vivant, comme en témoignent de nombreux travaux et même la fondation d'un nouveau groupe de recherche à l'université de Michigan : CLAIR[4], *Computational Linguistics And Information Retrieval*. On notera ainsi plus particulièrement les thèses de [Loupy, 2000] avec de nombreuses expériences en anglais, et [Moreau, 2006] qui revisite les techniques de TAL en RI.

Le troisième constat est que les critiques font l'amalgame des approches linguistiques. Deux des trois stemmers intégrés dans [Harman, 1991] sont très brutaux ([Porter, 1980] et [Lovins, 1968]) et provoquent des chutes de précision quand ils mettent en relation des mots qui n'ont sémantiquement rien à voir (voir 5.6 pour quelques exemples). Le jugement est un peu moins sévère pour le troisième stemmer (« *S stemmer* ») parce que l'auteur estime qu'il peut correspondre à des attentes de l'utilisateur. Ce dernier stemmer permet notamment de mettre en relation le lien entre le singulier et le pluriel en anglais. Aucun de ces trois traitements ne peut être considéré comme un traitement linguistique très précis, et il est vrai que les traitements plus précis, à base de lexiques ou de règles, demandent des investissements considérables qui ne sont pas toujours à la portée des laboratoires ou des entreprises.

Le quatrième constat est que les méthodes d'évaluation sont critiquables, d'après les critiques des critiques. Ainsi, [Loupy et Crestan, 2004] postulent que les critères d'évaluation de précision et de rappel ne sont pas totalement adaptés pour évaluer l'apport de modules linguistiques dans le processus de la recherche d'information. D'après leurs résultats, l'intégration de ces modules mène à une amélioration des résultats de recherche en tête de liste, ce qui permet notamment un gain de temps à l'utilisateur.

Le cinquième constat est que l'analyse linguistique apporte des fonctionnalités avancées qui n'étaient pas imaginées de cette façon à l'époque où étaient formulées les critiques. Ces fonctionnalités permettent un gain de temps à l'utilisateur, ce qui est un critère ignoré dans les grandes campagnes d'évaluation.

Plus que de prendre position dans cette discussion sur l'utilisation de TAL dans la RI, il nous semble plus important de rappeler que la recherche d'information en soi est une matière difficile. Le dernier des neuf postulats de [Swanson, 1988] qui conclut les autres est parlant à

[3] Le projet MuchMore a développé un prototype de recherche d'information médicale interlingue utilisant plusieurs types de ressources. Voir le site du projet (http://muchmore.dfki.de/) pour plus d'informations.
[4] Voir le site du laboratoire : http://tangra.si.umich.edu/clair

ce sujet : l'indexation cohérente, efficace et complètement automatique est impossible[5]. Nous en concluons que plusieurs approches se défendent et qu'il n'y a sans doute pas de bonne réponse générale.

2.1.2 Un analyseur linguistique au cœur du moteur

Le modèle de RI mis en œuvre dans le moteur de recherche de Sinequa est la combinaison d'un modèle statistique et d'un modèle vectoriel. Les calculs des deux modèles se font sur des objets linguistiques provenant d'une série de traitements issus du domaine du TAL. Ces objets linguistiques sont le résultat de l'analyse accomplie par l'analyseur linguistique qui est au cœur du moteur de recherche de Sinequa. Le rôle central de cet analyseur dans l'architecture du système est illustré dans la figure 3. En effet, c'est le même module qui, en appliquant des ressources linguistiques différentes, analyse les documents du corpus et les requêtes des utilisateurs. Dans le premier cas, les analyses obtenues servent à construire l'index. Il s'agit des informations linguistiques qui sont le résultat des divers traitements linguistiques. Dans l'autre cas, l'analyse sert à obtenir les informations linguistiques nécessaires pour transformer la requête initiale en une requête interne capable d'interroger l'index. La comparaison telle qu'illustrée dans la figure 2 est alors possible. L'interrogation se fait dans une base SQL enrichie.

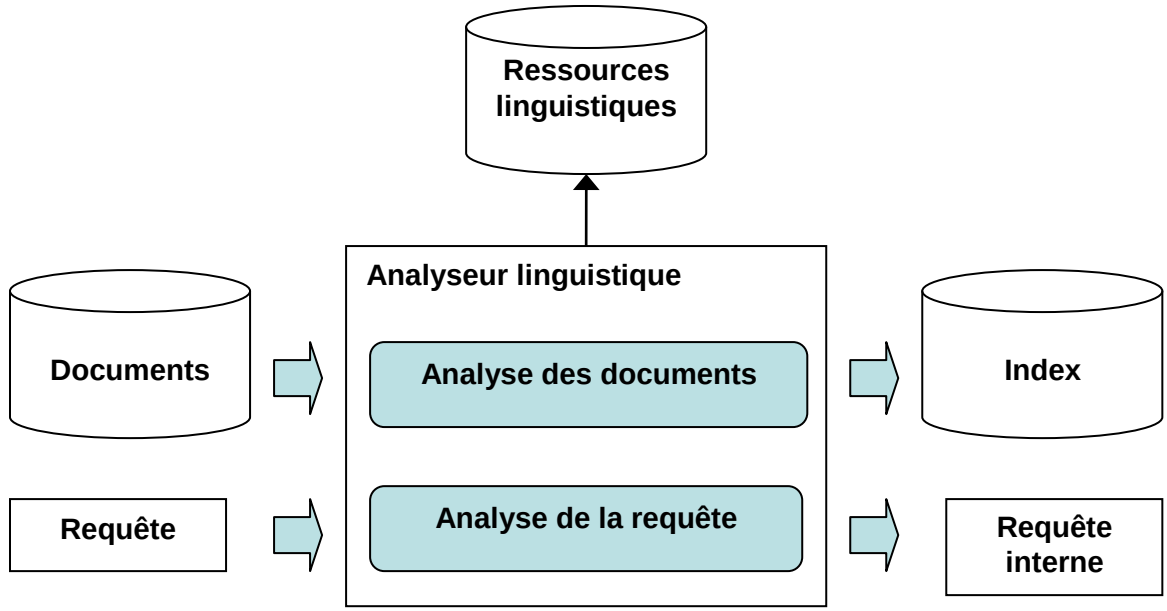


Figure 3 : Le rôle de l'analyseur linguistique dans le moteur de recherche de Sinequa

L'analyseur linguistique est un outil qui exécute différents traitements. La plupart d'entre eux mobilisent un certain nombre de ressources linguistiques, qui sont gérées par des linguistes. Selon le contexte de déploiement du moteur, des ressources adaptées ou spécialisées peuvent être branchées, et d'autres ressources peuvent être débranchées. La partie 3 et le chapitre 6 détaillent les différents traitements linguistiques et les ressources qu'ils mobilisent.

---

[5] "*In sum, the first eight postulates imply that consistently effective fully automatic indexing and retrieval is not possible. The conceptual problems of IR – the problems of meaning – are no less profound than thinking or any other forms of intelligent behaviour.*" (p. 96)

2.1.3 Une interface interactive motorisée par des techniques de TAL

Le calcul de la pertinence est un processus opaque pour l'utilisateur, et une bonne partie des informations issues de l'analyse linguistique reste plus ou moins cachée pour l'utilisateur. D'autres informations sont néanmoins directement exposées dans l'interface, comme les fonctionnalités de navigation rendues possibles par les analyses linguistiques.

Les recherches en navigation ont été motivées par le constat que l'évolution des performances des moteurs de recherche était arrivée à un pallier à la fin des années 90. Comme nous pouvons voir sur la figure 4, prise de [Harman, 2000], les systèmes ont connu de fortes progressions durant les campagnes successives de TREC, mais l'évolution a stagné à partir de 1996, indépendamment de la tâche à accomplir.

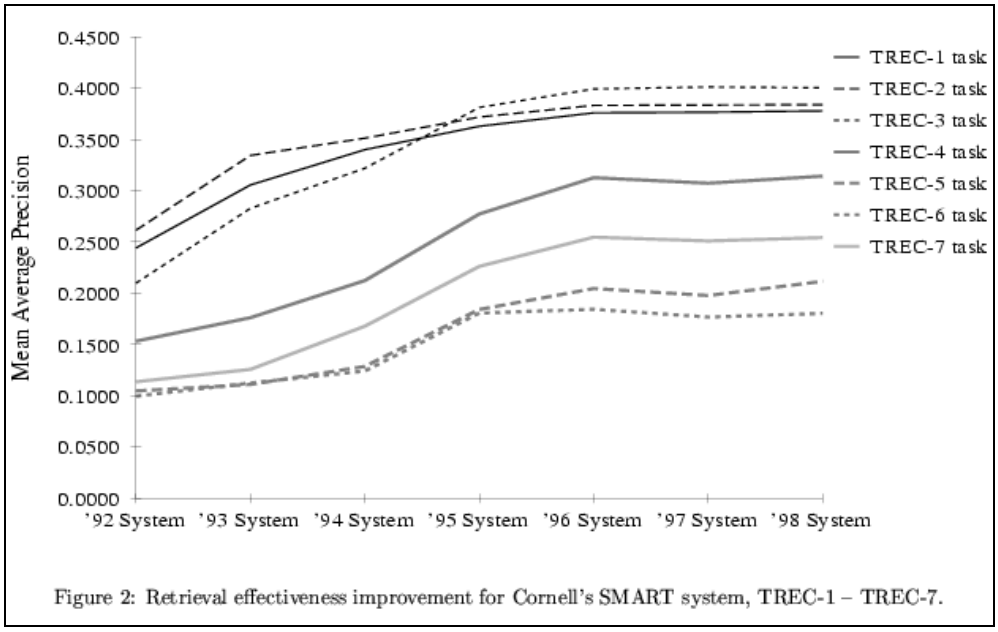


Figure 4 : Evolution de la performance des moteurs de recherche d'après les campagnes successives de TREC [Harman, 2000]

Fort de ce constat, Sinequa a mis en place au début des années 2000 une technologie de détection d'entités par automate. Elle permet de classifier des mots ou des suites de mots selon le contexte défini dans l'automate. Les exemples fonctionnellement les plus proches de cette technologie sont sans doute Intex ([Silberztein, 1993]) et Unitex ([Paumier, 2002]), implémentées d'après les idées de Maurice Gross ([Gross, 1989]). Sinequa a ainsi implémenté une propre version d'une technologie bien connue et répandue dans le domaine du TAL comme nous pouvons le lire dans le livre édité par [Roche et Shabes, 1997].

Dans l'interface du moteur de recherche de Sinequa, les entités détectées sont affichées dans les documents, mais aussi à coté de la liste des résultats dans des pavés affichant les entités les plus pertinentes. Un exemple est donné dans la figure 6 qui illustre la page de résultats et les différents pavés de navigation pour la requête *suite financière*, sur un corpus de magazine journalistique d'environ un an (du 3/4/08 au 30/4/09), totalisant 11 741 articles. Au cours de sa recherche, l'utilisateur peut peaufiner sa requête par un simple clic sur l'une des entités extraites. Aujourd'hui le produit affiche en standard les groupes nominaux courts (encore appelés *concepts*), les noms de personnes, géographiques et d'entreprises. Ces entités étant extraites des documents de la liste des réponses, l'utilisateur s'engage dans un processus de filtrage itératif en cliquant sur les entités. A notre connaissance, la navigation par affinage de

la requête en utilisant des termes simples avait été exprimée et expérimentée une première fois avec grand succès dans [Harman, 1988]. La méthode de détection des termes et le calcul de leur pertinence sont bien sûr totalement différents.

La navigation ne se limite pas au filtrage. Elle donne aussi de nouvelles idées à l'utilisateur qui peine à formuler sa requête, ou propose même une navigation spontanée à partir de la page d'accueil sans qu'il ait à poser une requête. Ce faisant, le moteur de recherche de Sinequa intervient pleinement sur le processus cognitif de l'utilisateur, instaurant une interactivité simple et aisée avec lui en facilitant sa recherche d'information. Comme l'illustre la flèche mauve dans la figure 5, c'est à partir des résultats obtenus par le moteur de recherche que l'utilisateur reformule sa requête ou qu'il revient sur le problème d'information initial, ce qui entraîne dans tous les cas la formulation d'une nouvelle requête. On souhaite en général que cette boucle soit aussi courte que possible, mais rien ne garantit que l'information cherchée soit présente dans la base.

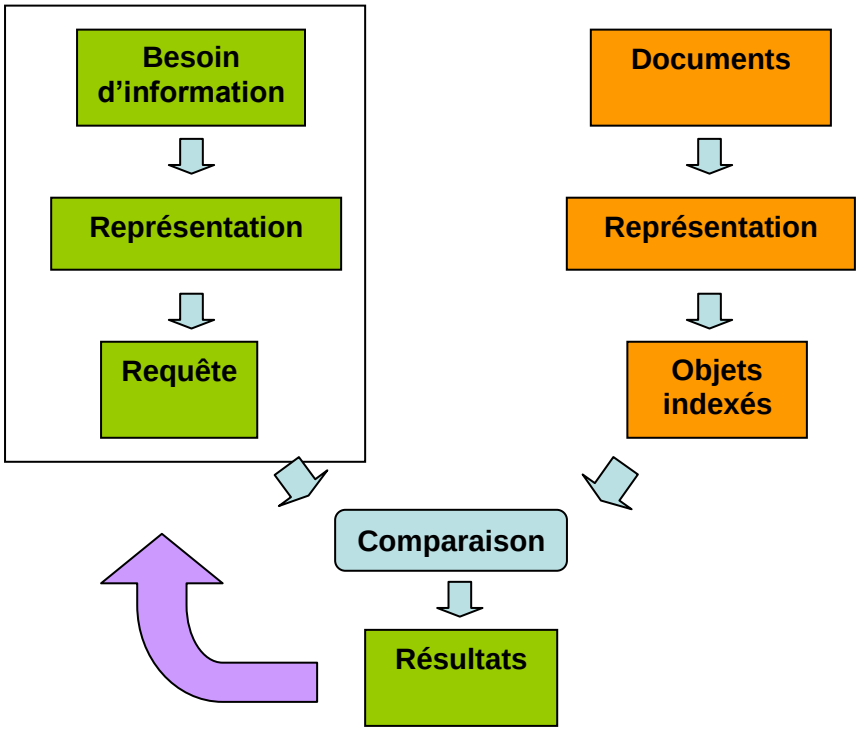


Figure 5 : Facilitation du processus de recherche à partir des résultats

L'efficacité de cette incursion dans le processus cognitif de l'utilisateur se vérifie dans [Crestan et Loupy, 2004]. Les auteurs ont évalué six interfaces plus ou moins riches en fonctionnalités sur un panel de six utilisateurs avec un jeu de 18 requêtes. Les résultats sont impressionnants. Les pavés de navigation apportent le gain de temps le plus important à l'utilisateur : par rapport à une interface classique, le gain de temps est de 25% pour trouver le premier document pertinent. Le gain monte jusqu'à 50% en utilisant l'interface fonctionnellement la plus riche. L'intégration de ces pavés dans l'interface de recherche est illustrée dans la figure 6. L'ensemble de mots coloriés en haut à droite n'est autre qu'une représentation alternative des « concepts associés », cette fois-ci sous forme de nuage.

L'apport de l'analyse linguistique semble clair dans le moteur de recherche de Sinequa. Elle contribue activement au succès du moteur à travers les fonctionnalités de haut niveau qu'elle rend possible. Tout doit donc être fait pour garantir la qualité de l'analyse linguistique qui est à l'origine de ces fonctionnalités.

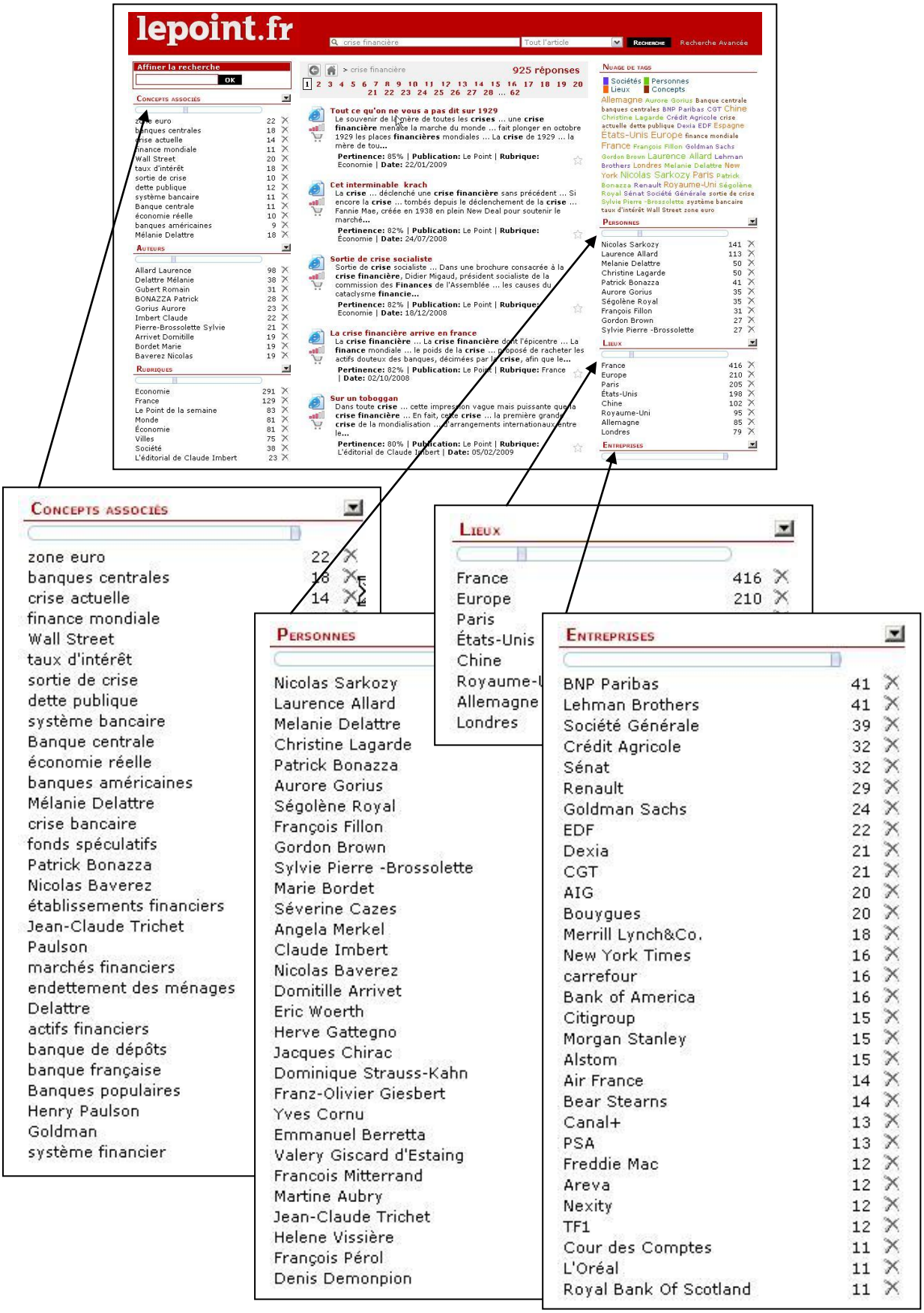


Figure 6 : Exemple de navigation pour la requête *crise financière*, fenêtre principale

2.2 Une analyse linguistique avancée

A l'origine d'un bon nombre de fonctionnalités qu'offre le moteur de recherche de Sinequa se trouvent des techniques de TAL avancées qui combinent des règles linguistiques et des statistiques. Le choix de Sinequa d'intégrer ces techniques dans son produit n'est pas seulement une conséquence logique de l'historique du produit ou de la société[6], il s'agit d'un vrai positionnement dans le paysage de la recherche d'information.

Dans la suite, nous présentons des fonctionnalités qui sont issues du TAL et qui sont disponibles dans la version actuelle du moteur dans la limite de l'existence des ressources linguistiques pour chaque langue. L'intérêt de chaque fonctionnalité pour la recherche d'information est explicité. Ces fonctionnalités reposent sur des traitements exploitant des ressources gérées par des linguistes-informaticiens à Sinequa.

2.2.1 Désambiguïsation lexicale

La désambiguïsation lexicale est l'opération qui consiste à attribuer une catégorie grammaticale à chaque mot dans le but de désambiguïser s'il y en a plusieurs possibles. Cela est important pour la levée d'ambiguïté sémantique en cas d'homographie et pour le filtrage de mots outils. Le but de la désambiguïsation lexicale est, en intervenant sur la requête et dans les documents, de réduire le bruit que peut provoquer l'ambiguïté grammaticale des mots.

Les homographes sont des mots qui ont la même orthographe mais portent des significations différentes, comme par exemple un *livre* et une *livre*, ce dernier mot étant triplement homographe comme mesure de poids, monnaie anglaise et comme verbe. Ces mots sont assez problématiques en RI, mais une certaine catégorie d'entre eux peut être traitée par la désambiguïsation lexicale. Si les homographes ont des catégories grammaticales différentes, comme c'est le cas pour l'ambiguïté entre le nom et le verbe. Il en va de même pour les mots *est* et *avions* en français, pour lesquels nous pouvons lever l'ambiguïté sémantique en levant l'ambiguïté grammaticale.

L'ambiguïté des mots grammaticaux est un phénomène assez fréquent dans les différentes langues. En néerlandais par exemple, le déterminant possessif *mijn* (mon/ma) est homographe du nom *mijn* (mine). Comme si cela ne suffisait pas le pluriel de ce même nom correspond à une ancienne forme accusative et dative du même déterminant : *mijnen*. Cette forme du déterminant est considérée comme archaïque ou dialectale, sauf dans une quinzaine d'expressions figées comme *te mijnen bate* (pour moi), *te mijnen behoeve* (à mon profit), *te mijnen gerieve* (pour mon confort), *te mijnen huize* (chez moi), etc. Un autre exemple en néerlandais concerne les noms de deux communes en Belgique, *Komen* (Comines) et *Voeren* (Fouronnes). Ces deux noms propres sont également des verbes à l'infinitif très courants : respectivement *venir* et *guider/transporter*. La capitalisation n'est pas toujours décisive, car il existe des constructions syntaxiques en néerlandais qui placent les infinitifs en début de phrase.

L'identification de la catégorie grammaticale donne aussi la possibilité d'ignorer certaines catégories de mots dans le processus de recherche. On appelle ces mots les *mots outils*, les *mots grammaticaux* ou encore les *mots vides*, sous-entendu vides de sens. Grammaticalement

---

[6] Avant sa fondation en 2000, Sinequa s'appelait CORA, une société d'ingénierie linguistique. L'un de ses projets de recherche donna naissance à *Darwin*, le lointain prédécesseur du moteur de recherche actuel. Les expériences de *Darwin*, système documentaire pour la presse, ont été capitalisées dans *Intuition*, le moteur de recherche à l'origine de la technologie actuelle qui a été rebaptisée *Sinequa Engine* en 2006.

il s'agit des déterminants, des pronoms, des conjonctions, des auxiliaires, etc. Par opposition les *mots pleins* sont typiquement les noms, les verbes, les adjectifs et les adverbes.

Les mots outils sont les mots de la langue les plus fréquemment utilisés, ce qui fait qu'un calcul statistique peut également permettre de les détecter, mais de façon très imprécise. Les fréquences indiquées dans le Tableau 1 sont calculées à partir des chiffres fréquentiels présents dans Lexique 3.5 ([New et al., 2004], [New et al., 2007]) qui sont calculés sur du corpus littéraire. On constate qu'une approche statistique qui tient uniquement compte de la fréquence ne sait pas faire la distinction de façon nette, car il n'existe pas de vraie frontière entre les mots vides et les mots pleins. Le premier mot plein de la liste apparaît à la position 97 avec le mot *yeux*. Les suivants sont aux positions 102 à 104 (*tête*, *dire*, et *homme*), suivi de *vie* à 106, *jour* à 108. Ces premiers mots pleins sont intercalés avec des groupes de mots grammaticaux et la densité des mots pleins va rapidement croître. On ne sait pas très bien non plus quel serait le statut des prépositions dans la liste, puisque d'après les fréquences on pourrait les considérer aussi bien comme des mots vides que pleins. La frontière entre mot vide et mot plein ne peut donc être bien faite par une mesure de fréquence uniquement.

Position	Forme	Fréquence
1	de	38 934
2	je	25 983
3	la	24 896
4	et	20 882
5	le	20 735
6	à	19 250
7	l'	16 422
8	il	15 832
9	tu	14 662
10	un	13 947
11	vous	13 590
12	d'	11 876
13	en	11 405
14	les	9961
15	une	9831
16	ça	8972
17	pas	8922
18	on	8586
19	dans	8296
20	est	7984
…		

Position	Forme	Fréquence
96	après	959
97	yeux	956
98	étaient	928
99	avais	919
100	leurs	894
101	entre	893
102	tête	861
103	dire	857
104	homme	852
105	aux	842
106	vie	835
107	avoir	829
108	jour	826
109	toute	802
110	ont	794
111	trop	790
112	devant	788
113	juste	783
114	petit	769
115	avant	737
…		

Tableau 1 : Fréquences cumulées des formes calculé à partir de Lexique3 (corpus livres)

L'approche combinée entre désambiguïsation lexicale et filtrage de mots outils prend le contre-pied d'un filtrage aveugle par une liste de mots vides, qui se fait en ignorant la catégorie grammaticale. Ainsi, dans notre approche, si on choisissait de ne pas indexer le verbe être, le mot *est* est seulement filtré en tant que verbe, mais jamais comme nom. Evidemment, même pour ce type de filtrage fin tel que nous l'appliquons il existe des contre-exemples comme *The Who* en anglais ou le nom du film *Etre et avoir* en français : il s'agit de noms propres qu'il faut identifier pour éviter qu'ils ne deviennent introuvables.

2.2.2 Lemmatisation

Un même mot se manifeste dans les langues flexionnelles sous différentes formes. Celles-ci portent des marques morphologiques qui sont définies par les lois flexionnelles de la langue en question. L'ensemble de ces formes constitue le *paradigme de flexion* ou *paradigme flexionnel* du mot, et la forme choisie par convention dans cet ensemble est le lemme du mot. Le lemme est donc une forme canonique choisie dans le paradigme qui représente l'ensemble des mots qui sont liés entre eux par la flexion.

Le nombre d'entrées dans le paradigme varie fortement selon les langues. Il est par exemple bien plus grand dans les langues à cas à cause du marquage des cas sur les formes. Le lemme en allemand et en russe représente donc bien plus de formes qu'en anglais ou en français, comme nous pouvons voir dans les exemples suivants.

Exemple FR : ambassadeur

Forme fléchie	*description*
ambassadeur	singulier
ambassadeurs	pluriel

Exemple EN : *ambassador* (ambassadeur)

Forme fléchie	*description*
ambassador	singulier
ambassadors	pluriel

Exemple DE : *Ambassadeur* (ambassadeur)

Forme fléchie	*description*
Ambassadeur	nominatif, accusatif, datif singulier
Ambassadeure	nominatif, accusatif, datif pluriel
Ambassadeuren	datif pluriel
Ambassadeures	génitif singulier
Ambassadeurs	génitif singulier (forme alternative)

Exemple RU : Представитель (représentant)

Forme fléchie	*description*
представитель	nominatif singulier
представители	nominatif pluriel
представителя	accusatif, génitif singulier
представителей	accusatif, génitif pluriel
представителю	datif singulier
представителям	datif pluriel
представителем	instrumental singulier
представителями	instrumental pluriel
представителе	locatif singulier
представителях	locatif pluriel

Dans les exemples ci-dessus, la forme féminine (*ambassadrice*) n'est pas comprise dans le lemme, car nous avons choisi de ne pas laisser varier le genre à l'intérieur d'un paradigme de flexion sauf exception linguistique.

Pour la recherche d'information, le phénomène de la flexion multiplie les objets analysés. Une normalisation par lemme est tout d'abord une façon de réduire considérablement le nombre d'objets pour augmenter l'efficacité des calculs statistiques de pertinence d'une part.

C'est également une extension de la requête faite au profit du confort de l'utilisateur : en indexant les lemmes en plus des mots-formes, on peut étendre la recherche sur toutes les formes fléchies du même paradigme. Si cela semble important pour toutes les langues flexionnelles, cela l'est d'autant plus pour les langues à cas, le nombre de mots-formes étant plus ou moins proportionnel au nombre de cas. Nous pensons bien sûr à l'allemand, au polonais, au russe, au finnois, etc.

Dans certains cas, il faut néanmoins savoir ne pas lemmatiser. Certains noms propres sont ambigus avec des mots communs. Ainsi faut-il savoir les détecter pour éviter une mauvaise lemmatisation. Si on ne détecte pas que *Christophe Belle* est un nom de personne, on lemmatiserait *Belle* en *beau*. D'où l'importance d'une bonne interaction entre les différents traitements.

2.2.3 Gestion de l'affixation grammaticale

Dans certaines langues, des suffixes grammaticaux se collent aux mots, remplissant la fonction d'un mot grammatical comme un déterminant ou une préposition. En suédois par exemple, la détermination d'un nom s'exprime par l'ajout d'un suffixe *–en* à la fin de ce nom.

SV	administration	administration + ∅	(administration)
	administration**en**	administration + en	(**l'**administration)

Dans quasiment toutes les langues nordiques, mais également en anglais, il existe le *-s* possessif qui se suffixe à la fin d'un nom pour exprimer un lien de possession avec un autre nom.

DE	Calvin Swift**s** Humor	Swift+s	(l'humour de Calvin Swift)
EN	Mister Jack**'s** bones	Jack+'s	(les os de Mister Jack)
NL	Schanullek**s** bontjas	Schanulleke+s	(le manteau de fourrure de S.)

En néerlandais et en allemand, l'adjectif qualificatif qui suit respectivement les mots *iets* et *etwas* reçoit également un suffixe *–s* ou *-es*.

NL	iets grappig**s**	grappig+s	(quelque chose de drôle)
DE	etwas komisch**es**	komisch+es	(quelque chose de bizarre)

En portugais, nous avons rencontré le cas de l'insertion de pronoms qui peut même transformer la forme du radical comme dans le second exemple.

PT	inscrever-nos-emos		(nous nous inscrirons)
	lavá-las-ão	lavar + as + ão	(ils les laveront)

Nous appellerons langues *agglutinantes* les langues dans lesquelles l'affixation grammaticale est une caractéristique principale du système morphosyntaxique. La seule langue sur laquelle nous avons pu travailler qui entre dans cette catégorie est le finnois. D'autres langues qui ont les mêmes caractéristiques sont l'estonien, le turc et le hongrois.

FI talossani talo (*maison*) + ssa (*dans*) + ni (*ma*) = (*dans ma maison*)

Quel que soit le degré d'affixation de la langue, les formes affixées doivent être identifiées et décomposées pour assurer une lemmatisation correcte. En effet, si les mots sont indexés tel

quels, une recherche sur un mot ne ramènera pas tous ses formes fléchies, ce qui serait une cause de silence. Ainsi en suédois, si on ne traitait pas l'affixation, la requête « administration » ignorerait les occurrences de « administrationen » alors que *administration* n'apparaît pas forcément dans les documents.

2.2.4 Identification et décomposition des mots composés

En morphologie, la composition désigne l'opération de former un mot composé à partir de plusieurs mots autonomes. Les auteurs de [Riegel et al., 1999] statuent dans le chapitre sur les *mots complexes* (chap. XVII, 3.2 et 3.5) que « en principe, tous les éléments entrant dans la formation d'un mot composé sont des unités lexicales autonomes par ailleurs […] sont donc eux-mêmes des mots de la langue, possédant leur autonomie syntaxique et appartenant à une catégorie grammaticale déterminée. » Les auteurs signalent deux problèmes pour cette définition traditionnelle : d'une part les composants des composés dits *savants* (comme *polymorphe*) ne sont pas forcément autonomes, et d'autre part il faut savoir distinguer le mot composé du simple groupe de mots (*chaise longue* par rapport à *chaise confortable)*. Plus loin, ils indiquent que « La composition savante peut être associée à la dérivation affixale. », mais le problème de la reconnaissance n'est pas plus développé (voir 2.2.5). Celui-ci est d'autant plus difficile que « l'orthographe des mots composés ne saurait suffire pour identifier [les mots composés] comme tels. » : il y a soudure graphique (*portefeuille, clairvoyant*), liaison par trait d'union (*chef-lieu, sourd-muet*) ou séparation par un blanc (*eau de vie, garde champêtre, à cause de*).

Ces trois façons d'écrire les mots composés se retrouvent dans les autres langues. Les règles orthographiques étant différentes selon les langues, c'est en général une des trois procédés qui est privilégié. Les règles orthographiques qui régissent la composition peuvent être assez libres ou mal connues du public, laissant la voie libre à l'existence de plusieurs variantes de la même composition. La réalité linguistique est souvent bien différente des idées reçues sur les langues.

En règle générale, les mots composés en anglais sont formés avec des espaces. On constate néanmoins que les mots sont parfois concaténés, phénomène qui devient même très courant en langue spécialisée, notamment médicale. En allemand et en néerlandais, la concaténation est la règle, mais les règles prévoient qu'on peut et dans certains cas qu'on doit, mettre des tirets pour augmenter la lisibilité du mot. Dans ces deux langues les espaces sont donc en théorie prohibés. Sauf qu'en néerlandais, sous l'influence de l'anglais, l'utilisation de l'espace est devenue tellement courante qu'il existe même un site[7] militant contre cette erreur orthographique où l'on peut signaler, photo à l'appui, les utilisations abusives. Nous appellerons les langues dans lesquelles la concaténation est la règle, les langues compositionnelles.

Voici quelques exemples dans ces différentes langues que nous avons vérifiés sur internet en utilisant des moteurs de recherche[8].

EN *database*, *data base* (base de données) : les deux variantes existent ; le Wikipedia anglais fait même une redirection de l'un à l'autre.

[7] Le site (http://www.spatiegebruik.nl/) s'appelle *Signalering Onjuist Spatiegebruik* (SOS), ce qui signifie *Signalement de l'utilisation injuste de l'espace*.

[8] Si les chiffres ne sont pas exacts et qu'il faut faire attention à leur interprétation (cf. [Kilgarriff, 2007]), ils donnent quand même des ordres de grandeur (tout à fait relatifs).

NL *monsterscore* : *monster* | *score* (score monstre) : la concaténation est obligatoire, mais on trouve aussi *monster score* dans les textes écrits.

NL *ski-jas* (veste de ski) : le tiret est obligatoire pour éviter la confusion avec le digraphe *ij*, mais *skijas* est malgré tout communément écrit, contre les règles orthographiques en vigueur.

databaseontwikkelaar, database-ontwikkelaar : l'usage du tiret est facultatif parce que *eo* n'est pas un digramme en néerlandais ; on peut néanmoins mettre un tiret pour faciliter la lecture.

driedoornige stekelbaars (épinoche) : l'espace est usuel dans tous les noms désignant des animaux.

DE *Arbeitssuche* : Arbeit | Suche (recherche d'emploi): la concaténation est obligatoire et respectée ; on recense quelques cas avec espace ou tiret.

Mittelmeerunion, Mittelmeer-Union (Union pour la Méditerrannée) : l'usage du tiret est facultatif et les deux orthographes coexistent.

En français, les règles orthographiques ne sont pas non plus toujours respectées : l'utilisation du tiret prête souvent à confusion. Pour le mot *porte-avions*, nous avons ainsi constaté des *porte avions* et même des *porteavions*. Que les règles d'orthographe ne soient pas toujours très strictes, ni respectées ou bien connues, si bien que la composition est génératrice d'un grand nombre de variantes possibles, qu'il convient de mettre en relation pour la recherche d'information.

Les mots composés sont très intéressants en RI, car ils ont un sens propre qui n'est pas simplement l'addition de leurs composants. Il convient donc de ne pas trouver *pomme de terre* à partir d'une requête *pomme* ou inversement. Néanmoins, il faut savoir les identifier. Comme l'atteste [Gross, 1988], le figement des mots composés est variable : certains sont plus figés que d'autres. *Pomme de terre* est très figé, alors que *planche à voile, plante verte*, le sont beaucoup moins. *Table ronde* l'est parfois, et *table carrée* ne l'est pas. S'il y a des critères linguistiques qu'on peut appliquer, le recensement des mots composés reste un travail manuel de longue haleine. En recherche d'information les critères pour distinguer les mots composés sont donc avant tout sémantiques et non morphologiques.

Un problème quelque peu différent existe également dans les langues compositionnelles. En allemand, *Federbett* (couette) est composé de *Feder* (plume) et de *Bett* (lit). Tout néerlandophone décomposera *stekelbaars* (épinoche) en *stekel* (épine) et *baars* (perche), *waterhoofd* (hydrocéphale) en *water* (eau) et *hoofd* (tête). Si on estime que ces mots ont un sens qui n'est pas l'addition des composants, il faut les considérer comme des mots simples. Tout comme *pomme de terre*, les composants ne doivent pas être indexés séparément. Il existe bien sûr des concaténations qui entrent dans le schéma de la *table carrée*, c'est-à-dire qu'il faut les décomposer. *Meubelhandel* (commerce de meubles) est un bon exemple, qui peut se paraphraser avec les composants par *handel in meubels*. Des exemples en néerlandais de mots plus ou moins figés sont w*alvisvangst* (pêche à la baleine) ou *visvangst* (pêche), qui sont respectivement composés de *walvis* (baleine) et de *vis* (poisson) plus *vangst* (capture). Pour ce type de mots, il est moins clair s'il faut les décomposer, et des arguments dans les deux sens peuvent être apportés.

Quand il y a des séparateurs ou des tirets, la décomposition ne pose pas de problème. La situation est différente quand les mots sont concaténés. Dans ce cas, il faut savoir décomposer à partir de règles et de lexique.

Dans les langues compositionnelles, la concaténation est à l'origine de ce que nous appelons la *factorisation*[9], un autre phénomène linguistique qu'il faut savoir traiter. Quand deux mots composés qui sont séparés par une virgule ou une conjonction et qui ont un même composant initial ou final, le composant en commun se remplace dans un des deux mots par un tiret pour éviter la répétition.

DE *Institut für Arbeitsmarkt- und Berufsforschung*
= Institut für Arbeitsmarkt**forschung** und **Berufsforschung**
(Institut pour l'étude du marché du travail et [de l'étude] du métier)

NL *vaktaal en -literatuur*
= vaktaal en vakliteratuur
(langue [professionnelle] et littérature professionnelle)

Par le biais de la morphologie et de la syntaxe on peut ajouter un comportement sémantique au moteur. La compréhension des mots composés augmente la précision et le rappel en même temps.

2.2.5 Identification et analyse des mots dérivés

Mots composés et mots dérivés ont en commun que leur construction est régie par des lois morphologiques très strictes[10]. La spécificité des mots dérivés est qu'au moins un des éléments n'est pas un élément autonome, mais un affixe (préfixe ou suffixe).

FR	*antichambre*	anti | chambre
	anticonstitutionnellement	anti | constit[uer] | ion | elle | ment
NL	*covoorzitterschap*	co | voorzit[ten] | ter | schap (coprésidence)

Le mot dérivé contient au moins un élément qui constitue le composant sémantique dont le sens sera modifié par les affixes, sauf dans la *composition savante*. Celle-ci est la concaténation de préfixes et de suffixes grecs ou latins, parfois sans autre base pour former un nouveau mot : *doryphore*, *télégraphe*, *agriculture*, *démocratie*, *génocide,* etc. Si ces mots nous semblent être une seule unité de sens, ce n'est pas le cas en grec, où ils sont bien perçus comme des mots construits.

EL	τηλέγραφος	|tilegrafos|	(télégraphe)
	τηλέ + γραφος	|tile| + |grafos|	

Comme la dérivation est un phénomène productif, elle donne lieu à la production de néologismes qui ne trouveront pas forcément leur chemin dans le dictionnaire mais qu'on rencontre tout de même dans les textes. Par exemple, depuis l'élection de Nicolas Sarkozy, on parle dans les médias de *sarkozysation* (des esprits, de l'audiovisuel français, de France Inter, etc.) ou encore du fait que la France se *sarkozyse*.

La reconnaissance de ces mots dans notre système est rendue possible par des règles et du lexique, le nombre d'affixes étant une liste fermée. Cette reconnaissance dynamique rend l'analyse moins dépendante des lexiques, et plus souple par rapport au traitement des néologismes, peu importe s'ils s'intégreront définitivement à la langue ou non. La décomposition ouvre la voie à une indexation paramétrable des composants ainsi qu'à la

[9] Nous n'avons pas trouvé de mot précis dans la littérature.
[10] [Riegel et al., 1999] les regroupent d'ailleurs sous le noms de mots construits, indiquant emprunter le terme à Danielle Corbin

lemmatisation. En effet, si on peut identifier et décomposer les mots dérivés, on peut également les lemmatiser et leur donner une catégorie grammaticale.

2.2.6 Normalisation orthographique

La normalisation orthographique est l'opération de de mettre en relation deux variantes orthographiques d'un même mot. Ce sont des mots qui ont la même description morphosyntaxique et sémantique et une phonétique identique ou presque, mais dont la graphie est différente de quelques lettres. Il se peut qu'une des variantes soit considérée comme l'orthographe standard, mais nous nous soucions de la réalité linguistique. On constate que les textes contiennent, pour des raisons multiples, de nombreuses variantes orthographiques qu'il convient de mettre en relation indépendamment du carcan des règles orthographiques. Pour la RI, elles doivent être traitées comme si elles faisaient partie du même paradigme.

Un premier type de variation est la faute de frappe. Elle est en général traitée par l'application d'un algorithme de similarité comme la distance de Levenshtein : les lettres du mot à corriger subissent des opérations de substitution jusqu'à ce qu'un mot semblable soit trouvé. Cette technique bien connue est traditionnellement utilisée pour pallier les erreurs orthographiques des requêtes utilisateurs et prend alors le nom de *recherche floue*. Les algorithmes mis en œuvre ne sont pas linguistiquement motivés et n'utilisent pas de ressources linguistiques. De ce fait, ils tombent en dehors du cadre de cette thèse.

Un deuxième type est caractérisé par la présence de diacritiques. L'absence d'un diacritique peut prêter à confusion entre les mots, comme pour les mots suivants :

FR maïs/mais, tache/tâche, révolté/révolte

S'il n'existe pas d'ambiguïté sur les formes une fois les diacritiques supprimés, la reconnaissance des mots devrait être assez relâchée pour prendre en compte le fait que les utilisateurs ne les écrivent pas toujours.

Le traitement est spécifique à chaque langue, car les diacritiques ont des fonctions différentes selon les langues. En néerlandais les diacritiques sont plutôt rares : ils donnent des indications phoniques et ne servent pas à discriminer des mots mais à améliorer leur lisibilité. Ainsi le tréma sert à indiquer une coupure phonétique syllabique :

NL *reeën* (pluriel de *ree*, cerf)

meeëten (partager le repas) : Als hij met ons meeëet (S'il mange avec nous)

En néerlandais, un accent aigu peut se mettre sur n'importe quelle voyelle tonique pour mettre un accent tonique. L'accent aigu ne différenciant jamais deux mots en néerlandais, il peut être ignoré à l'analyse des documents.

NL *Hárd !* (Dur !)

En russe c'est le tréma qui sert d'indicateur de l'accent tonique, mais il ne s'écrit pas systématiquement.

RU *актёр* ou *актер* (acteur)

Les diacritiques ne sont pas toujours de simples indicateurs d'accents, ils peuvent faire intégralement partie de la lettre. C'est le cas notamment pour le *å* et le *ä, ö, ü* qui, en suédois et en allemand respectivement, ne sont pas des indicateurs phoniques, mais des sons distincts. Ainsi, ils ne se confondent nullement avec leurs équivalents sans diacritique. Pour cette raison, on ne peut se fier à la table Unicode ([The Unicode Consortium, 2007]) qui décrit systématiquement la décomposition en *lettre + diacritique*. Cette description est réalisée pour

des raisons de codage et d'affichage, et non pour décrire le lien linguistique entre des caractères. Elle ne peut donc être utilisée (de manière détournée) que partiellement, pour les langues où les diacritiques n'ont pas beaucoup d'importance ou ne sont pas respectés.

Les trois lettres allemandes *ä, ö* et *ü* sont interchangeables avec les digrammes respectifs *ae*, *oe* et *ue*, créant des variantes orthographiques :

DE *hören = hoeren* (écouter)

Ces digrammes s'écrivent notamment quand on écrit tout en majuscules, ainsi qu'en Suisse, où les lettres avec trémas ne sont pas usuelles. La lettre *ß* s'écrit *SS* en majuscules. Pour le *ß*, il peut exister une confusion avec les règles en vigueur d'avant la réforme de l'orthographe allemande de 1996 qui a profondément changé les règles d'utilisation de *ss* et de *ß*, par exemple pour la conjonction *daß* devenue *dass*. Cette orthographe très controversée est définitivement entrée en vigueur dix ans plus tard, mais elle fait tellement peu l'unanimité que les éditions 2009 des deux dictionnaires principaux (Duden et Wahrig) ont des divergences dans l'écriture de certains mots.

En soi, cela n'a rien d'extraordinaire. De 1946 à 1995, l'orthographe néerlandaise avait une orthographe *préférentielle* et *progressive*, donc plusieurs orthographes admises pour certains sons. En voici deux exemples :

NL	*inse**k**t, inse**c**t*	(insecte)
	meeëten, mee-eten	(manger ensemble)
	DES-hormoon, deshormoon	(hormone DES)

Depuis 1995, *meeëten* s'écrit avec un tiret : *mee-eten*. Et depuis la même date, *insekt* est considéré comme une erreur, tandis qu'avant les deux formes étaient indifféremment acceptées. Le troisième exemple est une variation née lors de la dernière réforme qui date de 2005 : elle impose la 2^{e} graphie au détriment de la 1^{e}.

Et faut-il rappeler qu'en français, l'accent circonflexe n'est plus obligatoire – mais pas interdit non plus – dans la plupart des mots depuis les rectifications de l'orthographe du 6 juillet 1990 ? Ces variations systématiques qui sont prévues par le système orthographique constituent un troisième type de variation. Il est normal que le système orthographique évolue, mais cette évolution pose un vrai casse-tête en recherche d'information. Les différentes graphies coexistent, soit parce que l'auteur ignore la nouvelle graphie, soit parce que les textes traités datent d'avant les réformes.

Le quatrième type est celui des variantes qui ne sont pas systématiquement prévues par des règles orthographiques.

FR	plates-formes, plateformes	
NL	*tijd**s**verschil, tijdverschil*	(écart de temps)

On peut souvent généraliser ces cas parce que les auteurs appliquent les règles par analogie. On sera ainsi tenté d'écrire *platebande* au lieu de *plate-bande*. L'ajout du *s* dans l'exemple néerlandais est motivé par une différence en prononciation qui n'a pas d'autre incidence sur les caractéristiques du mot.

A l'origine de ce type de variation peut se trouver l'ignorance de l'utilisateur du moteur de recherche. Ne sachant pas comment écrire un mot, il l'écrit en appliquant des règles orthographiques sur la représentation phonétique qu'il a mémorisée. La différence entre les graphies peut être plus forte que dans les autres cas de variation, mais elles partagent l'image phonétique.

Un dernier type de variation est la translitération. Les théories Sens-Texte de Igor Mel'çuk sont connues en anglais comme celles de *Melcuk, Mel'cuk, Melchuk, Mel'chuk* et *Mel'čuk*. En français, on trouve la même pléthore d'orthographes sur le web. En russe, les noms d'entreprises étrangères sont tantôt écrits en cyrillique, tantôt en lettres latines. Il est important d'augmenter le rappel[11] en mettant en relation ses unités. Même s'il existe des règles officielles[12], elles ne sont pas toujours respectées.

Le traitement dépend donc fortement de la langue et, s'il est mal fait, peut provoquer d'énormes dégâts, comme on peut imaginer avec ces exemples en portugais et en français :

PT	maca	(brancard)
	maça	(masse, gourdin)
	maçã	(pomme)
FR	Bo	ville de 350.000 habitants en Sierra Leone
	le Bô	commune française de 100 habitants
	BO	abréviation de *Business Objects*, *Biarritz Olympique* et *Bulletin Officiel*

2.2.7 Prise en compte de liens sémantiques

Les mots sont liés entre eux par des liens sémantiques et ces liens peuvent être exploités pour enrichir la requête. Si l'apport de cela est discutable d'après la littérature scientifique, il l'est autant d'après les clients de Sinequa : certains clients apprécient la fonctionnalité tandis que d'autres préfèrent la désactiver.

Quatre types de liens sémantiques nous intéressent au premier plan. Le premier est celui de la dérivation qui couvre entre autres les nominalisations et les liens entre la forme masculine et féminine d'une profession ou d'un titre, mais aussi les mots dérivés dont nous avions discutés dans 2.2.5.

FR	fraiser – fraisage	(nominalisation)
	sarkozyser – sarkozysation	(nominalisation)
	auteur – auteure	(profession)

Les trois autres types de liens qui nous intéressent sont la synonymie, la variation orthographique et l'abréviation. Les vrais synonymes sont plutôt rares. Il s'agit de mots qui désignent la même chose et qui sont interchangeables si on fait abstraction du niveau de langage (ex. *vélo* et *bicyclette*, *car* et *autocar*). Avec variation orthographique, nous ne visons pas les erreurs orthographiques, mais les variantes bien établies. Il en existe pour de nombreux noms propres d'origine étrangère, comme par exemple la capitale *Rangoun, Rangoon, Yangon,* mais également pour des mots communs comme par exemple pour la langue turcique *ouïghour*, *ouïgour*, *ouigour*, *ouighour*. Le lien entre l'abréviation et sa forme développée n'est pas très évident du fait de la grande ambiguïté des abréviations : leur signification dépend du contexte. Le très connu sigle TGV en français peut ainsi donner *Train à Grande Vitesse*, mais aussi *Téquila Gin Vodka*, ou *Turbine Gaz-Vapeur*. Pour le sigle BO en exemple dans 2.2.6 nous avions mentionné *Business Objects*, *Biarritz Olympique* que *Bulletin Officiel,* mais comme pour toute abréviation il est possible d'en trouver beaucoup d'autres,

[11] Le rappel est obtenu en divisant le nombre de documents pertinents en réponse par le nombre de documents pertinents du corpus.
[12] Comme celles du Library of Congress : http://www.loc.gov/catdir/cpso/roman.html

comme Barack Obama, Bolivie, Back Office (produit Microsoft) ou Back Orifice (cheval de Troie)[13].

Si nous sommes capables de repérer ces liens, ils peuvent être paramétrables : mono ou bidirectionnel. Ces liens sont en général pris en compte au requêtage comme une expansion de la requête. Il est également possible de proposer une interactivité avec l'utilisateur comme nous l'avions expérimenté dans le cadre du projet ANR Vodel avec les ressources de Mémodata ([Dutoit et al., 2003]). Ces ressources contiennent des liens comme l'hyp(ér)onymie et l'antonymie.

2.2.8 Désambiguïsation sémantique

Les mots sont porteurs de sens, mais sont souvent ambigus. L'exemple classique est celui d'*avocat*, qui peut être à la fois la personne du barreau et le fruit. Une analyse sémantique du document peut lever ces ambiguïtés dans un contexte de recherche documentaire. Ainsi, selon que le texte contient d'autres termes juridiques ou d'autres noms d'aliments, il est possible de désambiguïser le mot *avocat*.

Le moteur de Sinequa le fait en attribuant à chaque document une valeur sémantique globale. Le calcul repose sur une ressource qui contient des descriptions sémantiques pour chaque mot référencé. La valeur est située dans un espace vectoriel de 800 dimensions qui représente l'univers, chaque dimension représentant une thématique. Dans le moteur de recherche, cette analyse est utilisée comme un paramètre dans le calcul de pertinence, ainsi que pour détecter des documents sémantiquement proches.

Les ressources à l'origine de ce traitement sont tellement riches qu'on peut les utiliser à d'autres contextes. Nous les avons ainsi utilisé pour réduire le nombre de traductions proposées pour une requête constituée de plusieurs mots ([Cailliau et al., 2008]).

2.2.9 Détection des entités et de leurs relations

Les entités sont des mots ou des suites de mots que nous identifions comme étant une entité sémantique à laquelle nous pouvons donner un label la caractérisant. Noms propres et références de dates sont mieux connus sous le nom d'entités nommées. En recherche d'information celles-ci jouent un rôle particulièrement important. Du fait de leur référence extralinguistique et non conceptuelle, elles ancrent le texte dans la réalité temporelle (références de date), géographique (noms géographiques) et situationnelle (noms de personne ou d'entreprise).

D'autres entités importantes pour le moteur de Sinequa sont les concepts, que nous appellerons des *groupes nominaux courts* : ce sont des groupes nominaux sans attachement syntaxique subordonné. Ce sont des éléments porteurs de sens très importants dans la construction syntaxique. Ils le sont d'autant plus quand ils contiennent plusieurs mots, car leur sémantique est plus précise et moins ambiguë que celle des mots simples.

Dans le moteur de recherche de Sinequa les groupes nominaux courts et les entités nommées sont essentiellement utilisés à des fins de navigation et d'aide à la lecture, et ils sont devenus des éléments essentiels de l'interface (voir 2.1.3).

---

[13] Le site http://www.acronymfinder.com/ n'en recense pas moins de 75 pour BO, la grande majorité en anglais, ce qui à notre avis est dû à la nature internationale du service rendu.

La proximité entre entités peut indiquer des relations entre elles et être exploitée pour construire un réseau de connaissance, ou encore pour créer des communautés virtuelles. Les entités sont ainsi les bases essentielles pour trouver une personne à travers une recherche de compétences.

La détection des entités nommées a des incidences sur d'autres analyses. Les noms propres sont des constantes et ne doivent pas être lemmatisés sous une autre forme. Ainsi le nom de famille de *Patrick Gateau* ne sera pas lemmatisé comme *gâteau*, et celui de *Christophe Belle* ne sera pas lemmatisé comme *beau*. Dans les langues compositionnelles, la décomposition peut être bloquée. Les noms propres ne seront pas découpés, même si ça se justifie morphologiquement. Ainsi la ville *Noordkaap* (*noord* = nord, *kaap* = cap) et le nom de famille *Oostland* (*oost* = est, *land* = pays) ne sont pas considérés comme des mots à décomposer, mais comme des mots simples.

2.2.10 Analyse multilingue

Le moteur de recherche de Sinequa est disponible dans 19 langues qui représentent une grande variété de familles linguistiques. Si les caractéristiques des langues romanes, germaniques, slaves, finno-ougriennes et asiatiques sont différentes, le cadre d'analyse linguistique pour toutes ces langues est le même : seules les ressources changent d'une langue à l'autre.

Si les documents peuvent être mono- ou multilingues, le moteur les traite de façon automatique. Il doit donc pouvoir identifier la ou les langues du document et les passages correspondants pour appliquer les bons traitements.

La recherche multilingue[14], au sens de poser une requête dans une langue et obtenir des documents dans une autre, est un champ de recherche à lui tout seul. Nous avons pu expérimenter le branchement de ressources externes, notamment en utilisant le service Senseagent de Mémodata, en combinaison avec une interface interactive qui donnait à l'utilisateur la possibilité de filtrer les bonnes traductions avant de continuer sa recherche ([Cailliau et al., 2008]). L'espace vectoriel à 800 dimensions dont nous parlions dans la section 2.2.8 utilisé pour l'analyse sémantique du document est le même pour toutes les langues. Comme chaque document est projeté indépendamment de sa langue dans le même espace vectoriel, il est possible d'identifier des documents proches de langues différentes. Le procédé étant totalement sémantique, les résultats ne paraissent pas toujours limpides à un utilisateur non averti : il s'attend à un traitement qui s'apparente à de la traduction. La solution parfaite n'étant pas encore trouvée, le système doit rester ouvert pour des solutions à venir.

2.2.11 Analyse structurelle du document

Selon la méthode déployée, la longueur du document peut jouer un rôle essentiel. Dans un environnement où les documents sont assez standardisés, comme dans un fonds documentaire d'articles journalistiques, la taille des documents ne pose pas vraiment un problème. Dans d'autres environnements, comme par exemple en entreprise, les documents peuvent être très courts (quelques lignes) comme très longs (quelques centaines de pages). Dans ce dernier cas, la question se pose du découpage du document en sous-documents, d'une part pour

[14] Le terme étant ambigu entre « interlingue » et « multi-monolingue », nous l'utilisons bien dans son acception habituelle en RI, qui est de passer d'une langue à une autre.

l'affichage des extraits pertinents en lieu et place du document complet, et d'autre part pour augmenter la pertinence de la recherche.

Ce découpage peut être thématique ou structurel. Dans le projet Infom@gic, notre partenaire EDF R&D a procédé à un découpage thématique de conversations téléphoniques ([Bozzi et al., 2009]) avant de mettre en œuvre la catégorisation sur ces mêmes segments. Ce découpage regroupe plusieurs tours de paroles thématiquement liés. Il était nécessaire car l'unité idéale pour la catégorisation n'était ni le document (=la conversation complète), ni les tours de parole qui étaient souvent très courts et sémantiquement pas très pertinents. Pour l'application du moteur de recherche nous sommes restés sur la conversation comme document, tout en adaptant l'interface du moteur aux propriétés particulières des documents et de l'audio ([Cailliau et Giraudel, 2008]). Le découpage thématique a produit des unités plus petites que le document, des segments, qui ont été catégorisés. Ces résultats ont été totalement intégrés dans le processus de recherche par une fonction de filtrage sur les thèmes attribués aux segments.

Dans le cadre du projet Textcoop l'équipe de recherche de Sinequa avait expérimenté une approche de segmentation du document par des éléments lexicaux assortis de règles. Le projet s'orientait sur des documents contenant des procédures comme des recettes de cuisine, des fiches de bricolage et des soluces vidéo. L'intégration dans le moteur de recherche et dans un système de questions-réponses a donné des résultats assez mitigés, avec une amélioration dans un nombre très restreint de cas ([Surcin, 2008]). Dans tous les cas, les règles et lexiques devant être redéveloppés pour chaque type de document, nous avons jugé cette approche trop coûteuse et trop peu générique pour un déploiement industriel.

Comme pour l'analyse multilingue, les recherches sont toujours en cours pour trouver la meilleure méthode pour le produit. Le système doit donc rester ouvert pour des solutions à venir.

2.3 Des techniques gourmandes en ressources linguistiques

Certains traitements participant à l'analyse linguistique sont gourmands en ressources linguistiques. L'implication d'autant de ressources est volontaire, car l'architecte logiciel aurait pu choisir des traitements qui reposent sur moins de ressources. Les modèles choisis sont essentiellement à base de règles linguistiques, qui ont l'avantage d'être consultables, explicables et corrigibles en cas d'erreur, ce qui n'est pas toujours le cas des méthodes à base d'apprentissage. Contrairement à ces derniers modèles, les règles sont plus flexibles car modifiables plus facilement.

Les traitements sont mis en place au fur et à mesure de l'évolution du produit. Si la communication entre les différents traitements est assurée dès leur enchaînement logiciel, les ressources sont en général mises en place indépendamment les unes des autres. Elles ne sont donc pas forcément au même format, et peuvent contenir des informations qui se recouvrent. Ainsi, un point de recouvrement entre un lexique sémantique et morphosyntaxique peut être la combinaison <*lemme-catégorie grammaticale*> qu'on trouve aussi bien dans l'entrée sémantique <*lemme-catégorie grammaticale-description sémantique*> que sur les entrées des formes morphosyntaxiques <*forme-lemme-catégorie grammaticale-traits grammaticaux*>. Les ressources, indépendamment de leur structure physique, sont donc liées entre elles par les informations qu'elles contiennent. Un autre exemple est donné par les grammaires, dont les règles exploitent la description morphosyntaxique des lexiques. Toute contrainte dans une

grammaire est ainsi profondément liée aux lexiques car elle utilise les catégories et les traits de cette description morphosyntaxique.

A cause de la complexité des traitements, mais surtout du grand nombre de langues, le nombre de ressources à gérer est très élevé. Les chiffres dans le tableau 2 donnent une idée du volume des ressources industrielles hors les corpus. Elles sont exclusivement sous format texte sauf pour les *ressources de traitement*, qui contiennent également quelques scripts et exécutables. Pour mieux faire sortir les chiffres, nous avons séparé les ressources servant à la validation des ressources linguistiques. Les *ressources d'exploitation* sont les ressources linguistiques sous forme binaire, alors que les *ressources d'exploitation secondaires* sont des fichiers de log et de contrôle générés pendant la compilation.

	Taille en Mo	Nb de fichiers	Nb de dossiers
Ressources linguistiques	1329,83	1830	285
Ressources de traitement	53,90	52	1
Ressources linguistiques de validation	1,84	162	154
Ressources d'exploitation	129,92	1141	42
Ressources d'exploitation secondaires	1097,36	500	29
Total	2610,11	3685	513

Tableau 2 : Volumétrie des ressources hors corpus (fév. 2009)

L'arborescence dans laquelle nous avons centralisé les corpus regroupe 102 corpus, totalisant plus de 3 millions de fichiers soit 48 Go dans des formats HTML, XML ou sans balisage, dont 5833 fichiers (546 Mo) sont zippés. Ces chiffres n'incluent pas les corpus achetés et stockés exclusivement sous forme de DVD.

Ces chiffres sont déjà parlants pour illustrer la masse de données à gérer qui concerne directement le produit. A ces données, il faudrait encore ajouter la documentation, les archives et les données concernant les projets de recherche pour avoir une image globale des données gérées par notre équipe à Sinequa. Néanmoins, afin de mieux interpréter ces chiffres et de comprendre pourquoi leurs grands volumes posent des problèmes de gestion, nous allons examiner ce que couvrent certaines ressources.

Nous les répétons ci-dessous dans le tableau 3 en les additionnant pour l'ensemble des langues et en y ajoutant les lexiques de noms propres (principalement des prénoms, des noms géographiques) et spécialisés. Les noms d'entreprise n'ont pas été pas comptabilisés, car ils sont produits à partir d'un fichier commun à toutes les langues. Ce tableau ne comptabilise pas non plus les informations sémantiques, ni les lexiques qui ne sont pas livrés en standard avec le produit de Sinequa.

	Nombre
Fichiers de lexique	134
Combinaisons lemme/cat	1 401 306
Mots-formes	10 427 876
Descriptions	16 809 812

Tableau 3 : Nombre d'informations gérées dans les lexiques morphosyntaxiques toutes langues confondues (fév. 2009)

Les mots-formes représentent une grande partie de la masse de données. A nombre de lemmes constants de 76 500, les mots-formes varient d'environ 143 000 en anglais et 406 500 en français, à 705 000 en portugais, plus d'un million en polonais et à 7 million en finnois. Nous présentons le détail cette projection dans l'Annexe I (tableau 34, p. 284) pour 12 langues non asiatiques[15].

Parmi les ressources linguistiques, nous comptons également les grammaires. Nous avons eu l'occasion d'en manipuler des différents : grammaire de désambiguïsation de la requête, modèle de langage pour la désambiguïsation du texte, grammaire de décomposition des mots composés dans les langues compositionnelles, grammaires d'extraction d'entités nommées, etc. Il en existe pour le produit, ainsi que pour les projets de recherche.

Leur existence et leur taille varient selon les caractéristiques des langues. Deux types existent sous un formalisme propre à Sinequa : les modèles de langage et les grammaires d'extraction d'entités. Les autres sont écrites dans des formalismes variés. Elles sont seulement quantifiables à travers le nombre d'entrées dans les ensembles de tests de non régression, sachant qu'il existe au moins un exemple par règle.

Les modèles de langage sont quantifiables en comptant le nombre de règles de désambiguïsation et le nombre d'entrées dans le lexique associé. Il en existe au moins un modèle générique par langue traitée. Des modèles de langage spécifiques peuvent également être appris, comme pour la grammaire de désambiguïsation des conversations développée dans le cadre d'Infom@gic ST2.31. Les moyennes pour les modèles de langage des douze langues[15] sur lesquelles nous avons pu travailler se chiffrent à près de 100 000 entrées lexicales et un peu plus de 400 règles.

Les grammaires d'extraction d'entités sont codées sous forme d'automates. Le nombre d'automates par langue dépend du niveau de commercialisation de la langue en question. Ainsi, dans les cinq plus grandes langues européennes[16] (FR, EN, DE, IT, ES), nous avons développé ou encadré l'extraction des entités de groupes nominaux courts et longs, et d'entités nommées de personnes, de noms géographiques et d'entreprises. Pour chaque langue, cela implique 9 automates dont 2 pour les groupes nominaux et 7 pour les 3 entités nommées. Pour cinq langues, le DA, NL, PL, PT, RU et le SV, seuls les groupes nominaux courts et longs ont été développés. En japonais (JA), seules les entités nommées existent, et en russe (RU) nous avons commencé le développement de la détection des noms de personnes et des entreprises.

En français, en anglais et dans une moindre mesure dans les autres langues, une panoplie d'autres entités existe mais ces entités ne sont pas déployées en standard dans le produit, comme la détection des plaques d'immatriculation, des numéros de téléphone ou encore des adresses. Egalement non déployés, il existe quelques grammaires en français de mise en relation d'entités.

Le nombre de grammaires gérées activement pour le produit de Sinequa est illustré dans le tableau suivant.

[15] DA, DE, EL, EN, ES, FR, IT, NL, PL, PT, RU, SV
[16] En nombre de locuteurs bien entendu.

Grammaires	Nb automates	Nb langues	Total
Groupes nominaux courts et longs	2	11	22
Les trois entités nommées	7	7	49
Autres			57
Total			128

Tableau 4 : Nombre de grammaires d'extraction d'entités

Dans certains projets de recherche, nous avons utilisé la même technologie pour détecter des entités spécifiques. Même si ces automates ne sont pas utilisés ou retravaillés activement par l'équipe linguistique, ils augmentent tout de même considérablement la masse de données présente.

Dans le cadre des projets de recherche, les automates ont souvent été générés à partir d'autres informations. Ainsi, pour le projet TSSRC, 56 automates (avec 72 lexiques) détectant des structures causales ont été générés à partir d'informations livrées par le LDI (Université de Paris 13). Dans le projet Blogoscopie, notre équipe a généré des automates à partir d'un lexique d'évaluations (contenant des opinions, des sentiments, etc.) qu'elle avait créé. Dans le projet Infom@gic ST2.31, une cartouche Temis élaborée par EDF a été traduite sous forme de cinq automates. Dans l'ensemble de ces cas, la gestion des sources est plus importante que celle des automates, qui peuvent être régénérés assez facilement. Egalement dans ces cas, la frontière entre lexique et grammaire est très réduite. Le lexique n'exprime pas de règles en soi, mais fournit toutes les informations interprétables par des règles minimales.

2.4 Contraintes de l'exploitation

L'exploitation des ressources linguistiques dans un cadre industriel impose des contraintes particulières à leur gestion.

Héritage du passé et continuité

Le produit, en l'occurrence le moteur de recherche, a un cycle de vie qui est propre à la vie d'un logiciel. Avec l'évolution du logiciel, de nouvelles versions sortent, et la maintenance est garantie jusqu'à deux versions d'écart avec la version actuelle. Nous estimons ce passage à environ 5 ans, ce qui veut dire qu'il faut pouvoir donner du support sur des versions du logiciel qui sont anciennes de 5 ans. Nous ne pouvons donc pas changer radicalement l'analyseur linguistique sans casser la rétrocompatibilité. En 2008 l'architecture du moteur a été totalement réécrite. Cela fait que deux versions de l'analyseur linguistique et des ressources coexistent. Cela ne facilite évidemment pas la gestion.

Un autre exemple qui illustre que nous devons assumer des choix techniques faits dans le passé et en tenir compte pendant une longue période est celui du passage à l'UTF-8 des lexiques. Pour homogénéiser l'encodage des lexiques morphosyntaxiques dans toutes les langues, nous voulions passer tous les lexiques au format l'UTF-8, qui est l'encodage choisi pour les dernières langues ajoutées. Or, cette mise à jour a été refusée pour la raison que la version de l'analyseur linguistique en production chez certains clients n'était pas compatible avec l'UTF-8. Cette mise à jour doit donc attendre que la maintenance pour ces versions du moteur soit arrêtée.

Robustesse

L'indexation et l'analyse linguistique sur laquelle elle repose sont des processus automatiques. Tout type de document doit pouvoir être analysé, quels que soit le format (XML, HTML, Word, Excel, PDF, etc.), le contenu (rapports, tableurs, mails…) et le niveau de langage (soutenu, relâché, etc.). Un analyseur robuste doit savoir se débrouiller dans les différents cas qui se présentent pour rendre des analyses correctes sans aucune intervention humaine. Cela veut dire que tous les traitements doivent être robustes, même ceux de relativement haut niveau comme l'extraction d'entités. Comme les résultats de ce dernier traitement sont directement visibles dans l'interface, aucune erreur n'est tolérée. Comme la stratégie ne prévoit qu'une même grammaire pour tout type de document, le silence est privilégié quand il y a des risques de bruit.

Rapidité de traitement

Si l'analyse linguistique rend possibles des fonctionnalités exclusives, elle ne doit pas augmenter de façon abusive les temps d'indexation. Un écart substantiel dans les temps d'indexation avec les produits concurrents se traduirait immédiatement dans un désavantage concurrentiel irrattrapable, quelles que soient les fonctionnalités supplémentaires. Dans les ressources, certaines règles doivent donc être respectées, telle la limitation de l'utilisation des expressions régulières dans les grammaires de détection des entités, qui sont particulièrement gourmandes en temps d'analyse. A la mise en place de l'analyse morphosyntaxique des nouvelles langues, nous avons veillé au bon équilibre entre données statiques (le lexique) et données dynamiques (résultat d'une analyse par règle) pour assurer de bonnes performances : seuls les phénomènes linguistiques peu récurrents mais très réguliers sont analysés par règle. Certaines technologies qui pourraient pourtant améliorer la qualité de l'analyse linguistique comme une analyse syntaxique profonde, ne peuvent être intégrées à l'état de l'art actuel pour cette même raison.

Maintenance

Tout problème signalé doit être corrigible. Cela fait partie du contrat avec le client. Pour cette raison, les modèles choisis pour les traitements sont des modèles à base de règles, et non pas des modèles statistiques qui sont difficiles à corriger. Cette contrainte façonne la conception des traitements mêmes et donc des ressources.

Reproductibilité

Tout comportement du moteur doit être reproductible. Il est tout à fait possible d'introduire des comportements aléatoires ou imprévus dans le système à travers les ressources. Il est par exemple possible que les résultats de la détection des entités varient selon la version de l'interpréteur si l'automate appliqué n'est pas déterministe. On veille donc spécialement à trouver toutes sources d'erreurs possibles et de mettre en place les procédures nécessaires pour les éviter.

Non régression

D'une version à une autre des ressources, mais aussi de l'analyseur, les résultats doivent être constants. Ils peuvent s'améliorer mais ne doivent surtout pas régresser. Pour chaque traitement nous avons mis en place un ensemble des tests de non régression qui est exécuté avant chaque livraison des ressources et avant l'intégration dans le produit. Ces procédures ont profondément modifié les méthodes de travail de l'équipe linguistique.

Coût

Le grand nombre de ressources est géré – pour des raisons économiques – par une équipe restreinte. La situation idéale serait que chaque langue soit gérée par une personne dont c'est sa langue maternelle, mais plusieurs problèmes s'y opposent. Le premier est d'ordre économique. Pour 19 langues, il faudrait autant de personnes, ce qui n'est économiquement pas viable[17], au moins dans le système économique actuel. Une aussi grande équipe demanderait par ailleurs des ressources supplémentaires en encadrement. Il existe heureusement des personnes plurilingues, ce qui peut réduire d'autant la taille de l'équipe. Le second problème est la pénurie sur le marché du travail : il n'est pas évident de trouver sur Paris les personnes qui ont exactement le profil cherché.

Les langues ne sont pas toutes égales devant le développement commercial. Selon la stratégie adoptée et les ventes, les langues au centre de l'attention peuvent varier. Celles qui ne sont pas au cœur de la stratégie ne sont pas délaissées, mais plutôt développées ou mises à jour ponctuellement, de façon opportuniste, quand l'occasion se présente.

Nombre de langues

Face à la concurrence, plus le nombre de langues traitées est grand, mieux c'est. Les autres moteurs n'hésitent pas à mettre en avant le nombre de langues qu'ils savent indexer. Or, il ne faut pas confondre analyse linguistique et indexation. Même le pire moteur de recherche peut indexer la quasi-totalité des langues du monde en tenant compte seulement des séparateurs comme l'espace et la ponctuation. Sans analyse linguistique spécifique, cela est bien inutile, comme le prouve l'exemple que nous avons mis en Annexe B, p. 253 : la requête *loutre* sur un moteur de recherche bien connu renvoie un site du gouvernement français en première réponse qui s'intitule « **L'outre**-mer ». L'apostrophe est le caractère par excellence qui s'interprète différemment dans les différentes langues. Ajouter une langue au catalogue de Sinequa ne se limite donc pas à une séparation des mots ou une reconnaissance de la langue, il s'agit de développer une analyse linguistique avancée, comme décrite dans le chapitre suivant.

Gestion unique

L'architecture de l'analyseur et le format des ressources sont les mêmes pour toutes les langues. Comme les langues ont des particularités linguistiques, cela veut dire que tous les mécanismes d'analyse sont disponibles dans toutes les langues, mêmes s'ils ne sont pas utilisés. Cette architecture est motivée par un souci de gestion industrielle. Avoir des briques logicielles spécifiques pour chaque langue compliquerait énormément l'intégration des ressources et leur gestion.

Généricité de l'analyse

Le modèle économique actuel de Sinequa est celui d'un éditeur de logiciel, et non celui d'une société de services. Le moteur doit pouvoir s'intégrer sans réglage linguistique spécifique au cœur du moteur. Contrairement à d'autres acteurs du TAL, nous ne créons pas de ressources par secteur ou par thématique abordée. Cela veut dire que la vision lexicale est celle de l'agrégation : dans les lexiques de Sinequa on trouve donc la langue générale et les langues de spécialité. Cela ne veut pas dire que dans l'avenir des modèles différents peuvent coexister et

[17] Quand nous avons développé le gros des langues, Sinequa comptait une douzaine de salariés. Milieu 2009, il en compte 35. L'équipe des linguistes en CDI a varié d'un (moi-même) à cinq.

s’appliquer selon le type de document analysé. Notre analyse en ce matière est fourni en Annexe J, p. 285.

Plusieurs distributions

Une distribution est un ensemble complet de ressources nécessaires au module d’analyse linguistique. Dans cet ensemble, des ressources peuvent être ajoutées, retirées, ou remplacées. Il peut s’agir de lexiques, mais aussi de grammaires de désambiguïsation morphosyntaxique, de détection d’entités, etc. Deux types de distributions existent : celles qui ont été adaptées aux besoins spécifiques de certains clients (mais c’est très rare et les adaptations sont minimales, ce qui prouve la généricité du moteur), ou bien des distributions relatives aux projets de recherche. Ces dernières distributions peuvent donner naissance à des déclinaisons futures du produit, comme par exemple une version du moteur spécifique pour analyser et indexer les appels dans les centres d’appel, suite du projet Infom@gic ST2.31.

2.5 Conclusion

Comme nous avons vu dans ce chapitre, la langue pose de nombreux défis à la recherche d’information, et l’intégration de techniques de TAL est une façon d’y répondre. C’est notamment le cas pour le moteur de recherche que nous avons étudié. Ces traitements permettent d’augmenter le nombre de fonctionnalités lors de la recherche, voire d’ajouter un brin d’intelligence. En contrepartie, à cause de leur richesse un nombre important de ressources linguistiques est nécessaire, et elles représentent une grande masse de données à gérer. La recherche d'information est donc le cadre applicatif des traitements et des ressources linguistiques sur lesquelles nous avons travaillé.

PARTIE 2

Etat de l'art et problématique

Chapitre 3

Les ressources linguistiques et leur gestion

De nombreuses techniques de TAL reposent sur des informations codées dans des ressources linguistiques. Si ces ressources sont indispensables au bon fonctionnement des traitements, leur développement peut demander un investissement considérable. L'acquisition et la mise à jour sont souvent considérées comme des opérations fastidieuses à cause du grand nombre d'informations et de leur complexité. L'investissement pour créer ces ressources pouvant être considérable, des organisations centralisatrices[18] ont été créées et des initiatives décentralisatrices[19] ont vu le jour pour les partager et les distribuer. Grâce à cette disponibilité croissante des ressources, celles-ci ont commencé à vivre leurs vies, déconnectées du but pour lequel elles ont été créées, à tel point que nous perdons parfois de vue leurs propriétés originales. Dans ce chapitre nous revenons sur ces propriétés, la vie des ressources et leur gestion ainsi que sur les outils de gestion existants.

---

[18] LDC (http://www.ldc.upenn.edu/) aux Etats-Unis ; ELRA (http://www.elra.info/) en Europe ; CNRTL (http://www.cnrtl.fr/) en France ; TST-Centrale : (http://www.inl.nl/tst-centrale) pour les Pays-Bas et la Flandre et le Suriname.

[19] http://www.mpi.nl/INTERA/ ; [Broeder et al., 2004] Ce projet a créé un répertoire européen de ressources linguistiques par la mise à disposition locale de métadonnées

3.1 Qu'est-ce une ressource linguistique ?

Le terme de *ressource linguistique* est souvent employé sans que sa définition soit vraiment posée. Avant de proposer la nôtre, voici deux définitions provenant de deux acteurs experts de la matière.

Définition trouvée sur le site d'ELDA[20] :

> « Les ressources linguistiques sont tous les types de données relatives à la langue, accessibles dans un format électronique, et utilisées pour le développement des systèmes de traitement de la parole et du texte dans des applications en technologies de l'information. »

Définition en provenance du manuel de GATE ([Cunningham et al., 2009]) :

> « Language Resource (LR): refers to data-only resources such as lexicons, corpora, thesauri or ontologies. Some LRs come with software (e.g. Wordnet has both a user query interface and C and Prolog APIs), but where this is only a means of accessing the underlying data we will still define such resources as LRs. »
>
> « Processing Resource (PR): refers to resources whose character is principally programmatic or algorithmic, such as lemmatisers, generators, translators, parsers or speech recognisers. For example, a part-of-speech tagger is best characterised by reference to the process it performs on text. PRs typically include LRs, e.g. a tagger often has a lexicon; a word sense disambiguator uses a dictionary or thesaurus. »

Ces deux définitions mettent l'accent sur des caractéristiques différentes, mais ne sont pas incompatibles. L'on retiendra de la définition d'ELDA que les ressources linguistiques contiennent des données relatives à la langue dans un format électronique et qu'elles servent à développer des systèmes de TAL.

La définition venant du manuel de GATE est autrement intéressante car elle définit *ressource linguistique* (données, déclaratif) par opposition à une *ressource de traitement* (programme, procédural), ce qui correspond aussi à la définition de [Habert, 2005] qui fait la distinction entre *instruments* (≈ PR) et *ressources* (≈ LR). Dans ces deux derniers cas, il est fait état d'une possible relation entre des ressources linguistiques et des programmes les exploitant directement.

A notre avis, ces définitions ne consacrent pas assez d'importance au sens original du mot *ressource*. Quelque chose n'est une ressource que s'il peut être exploité. Dans notre cas les ressources naissent de la séparation entre le code d'un programme et les connaissances qu'exploite ce programme. Cette externalisation des connaissances sous forme de données formatées fait que les ressources sont exploitables par tout programme qui vise le même traitement dans le même cadre d'application. Pour toute ressource il existe donc un programme, et par extension un traitement pour lequel la ressource a été créée et qui peut l'exploiter. Le lien entre le traitement et la ressource est très fort et compromet en général l'exploitation de la ressource par un autre traitement que celui pour lequel elle était prévue. La réutilisabilité de la ressource dépend donc de la distance entre le traitement pour lequel elle a été conçue et le traitement qui souhaite l'exploiter.

[20] http://www.elda.org/rubrique51.html; en ligne le 8/08/08

Nous proposons donc la définition suivante :

> Une *ressource linguistique* est un ensemble de données comportant des connaissances linguistiques exploitables par un traitement automatique en particulier.

Cette définition intègre le concept de *connaissances* tel que [Kayser, 1998], p. 30-31 l'exprime :

> « Les connaissances sont des données qui influencent le déroulement des processus. »

Dans notre définition, nous n'avons pas repris la notion d'*instrument* de Habert, car elle induit une certaine notion d'autonomie. Comme celle-ci n'existe pas toujours dans la chaîne de traitements que nous avons pu manipuler, nous lui avons préféré le terme plus abstrait de *traitement*. Nous pourrions considérer l'ensemble de la chaîne de traitement comme un instrument évolué, intégrant plusieurs traitements qui exploitent plusieurs ressources linguistiques.

La séparation entre programme informatique et données linguistiques est un choix pragmatique. Elle permet de confier la gestion du code informatique et des connaissances à ceux qui en maîtrisent séparément le sujet.

Nous souhaitons aussi faire la distinction entre ressources linguistiques et *ressources langagières*. Ce dernier terme est une traduction de l'anglais *language resources* et englobe des ressources bien plus vastes que les ressources linguistiques comme par exemple des dictionnaires électroniques ou papier, des exercices de langue, des correcteurs orthographiques, etc. Le terme de ressources langagières est donc beaucoup plus général et s'applique sur tout document (au sens large) ou logiciel qui a trait à la langue.

3.2 Trois types : corpus, lexiques, grammaires

Nous distinguons 3 types de ressources linguistiques : les corpus, les lexiques et les grammaires. Voici les définitions que nous proposons.

> Un *corpus* est un ensemble de documents textuels qui sont rassemblés selon des critères définis dans un certain but. Un corpus annoté comporte en plus des informations sur le contenu des documents.
>
> Un *lexique* est un ensemble d'unités textuelles ou sémantiques qui sont assorties d'informations les concernant.
>
> Une *grammaire* est un ensemble ou une liste de règles qui décrivent les opérations de transformation, d'enrichissement ou de suppression à effectuer sur les données qu'elles vont recevoir en entrée.

Elles peuvent toutes servir de ressources d'exploitation, de validation ou d'évaluation des traitements. L'*exploitation* est l'utilisation active lors du fonctionnement du système, la *validation* est la vérification du bon fonctionnement du système et l'*évaluation* est la mesure de la performance du système.

Ces trois types de ressources peuvent être très liés entre eux. Le lien peut être direct, une grammaire qui est apprise sur un corpus par exemple, ou bien indirect par les traitements qui les exploitent.

Les ressources ne sont pas égales devant la séparation entre programme et données. Le contenu des grammaires s'ajoute aux règles déjà encodées dans le programme, et est donc une

sorte d'extension des instructions du programme. Les lexiques au contraire servent de vocabulaire terminal pour les grammaires ou pour les instructions du programme. Les corpus, échantillon descriptif de la langue, ne remplissent aucun de ces deux rôles, mais servent à construire des règles ou du lexique.

La distance entre lexiques et grammaires n'est pas toujours très grande. Comme les lexiques fournissent le vocabulaire terminal aux règles, que celles-ci soient codées dans le programme ou dans les grammaires, il est possible de les voir comme des grammaires, à l'image du lexique-grammaire comme l'avait mis en œuvre Maurice Gross ou des grammaires d'unification[21] (LFG, HPSG, TAG, etc.). Plus les informations lexicales exploitées par un même traitement sont complexes, plus le lexique ressemble à une grammaire. Cette impression a été assez frappante pour les lexiques que nous avons manipulés dans le cadre des projets TSSRC, Blogoscopie et Infom@gic ST2.31, d'autant plus que nous avons transformé ces lexiques en grammaires sous forme d'automates avant de les exploiter. Nous considérons ces ressources néanmoins comme des lexiques car les descriptions lexicales décrivent ce qui est spécifique à chaque mot-forme ou lemme de l'entrée du lexique. Les règles codent ce qui est commun à toutes les entrées qui ont au moins partiellement la même description.

3.3 Organisation interne des ressources

L'organisation interne des ressources n'est pas normalisée, malgré plusieurs tentatives de mise en place d'un cadre commun comme nous le détaillerons dans le prochain chapitre. Les formats sont donc souvent propriétaires ou sont taillés pour l'application qui va les exploiter.

La situation est quelque peu différente pour les corpus à cause de l'intersection avec le monde documentaire qui a normalisé plus vite les descriptions des documents à des fins d'organisation. La Text Encoding Initiative (TEI)[22] s'est ainsi rapidement imposée pour la description de la structure des documents, mais laisse la main libre pour le formalisme interne de l'annotation linguistique. En 2001 est sortie la TEI/P4 qui impose XML comme langage d'encodage, qui remplace le SGML. La plupart des corpus sont codés en XML, suivant ou non les recommandations de la TEI. Si les corpus sont créés pour une application, le format est souvent propriétaire. Ceux qui sont créés à des fins plus générales comme la mise en place d'un corpus de référence sans but spécifique comme c'était le cas du BNC ([Burnard, 2000]) essaient néanmoins de suivre une certaine norme, voire de l'instituer si elle n'existe pas.

En ce qui concerne les lexiques, on rencontre deux types d'organisation : l'entrée est construite autour du lemme ou bien de la forme. Le lemme a toujours été l'accès principal au dictionnaire papier. C'est une organisation axée sur l'utilisateur, le lecteur. Les lexiques morphosyntaxiques sont des lexiques qui servent à des traitements de bas niveau, en confrontation directe avec le texte. Leur premier « utilisateur » est une application, ce qui fait qu'ils sont souvent construits autour du mot-forme[23] (comme Lexique3 [New et al., 2004] ou les lexiques morphosyntaxiques de Sinequa). Ce n'est pas le cas du lexique Morphalou[24], mais étant dérivé d'un dictionnaire papier, le TLF, ce n'est pas si étonnant. Par la suite le lemme a été remplacé par un identifiant comme référence de l'entrée pour séparer les entrées

[21] [Flickinger et al., 1985] décrit le passage de GPSG à HPSG : les règles syntaxiques sont passées de 350 à moins de 20 pour une même couverture en codant plus d'information linguistique dans le lexique.
[22] Voir aussi 4.1.1 (p. 64).
[23] Traduction de l'anglais *wordform*, le mot-forme a été défini par [Polguère, 2003] comme « un signe linguistique […] ayant une certaine autonomie de fonctionnement [et] une certaine cohésion interne.
[24] Disponible sur le site web du CNRTL (http://www.cntrl.fr).

ambiguës (par exemple les identifiants *boulanger_1* et *boulanger_2* pour respectivement le nom et le verbe). A notre avis, c'est moins l'origine qui est responsable de cette organisation que le type d'informations codées. Il est impossible d'inclure proprement des informations purement sémantiques dans un lexique morphosyntaxique organisé autour du mot-forme. Même sur la combinaison <lemme,catégorie grammaticale> cela est sujet à discussion. Dans tous les cas le lemme, héritage du monde lexicologique, garde son importance pour la gestion du lexique.

Les grammaires sous forme d'automates à états finis sont assez courantes : ils ont notamment été déployés dans Intex ([Silberztein, 1993]), Unitex ([Paumier, 2002]) et NooJ, le successeur d'Intex.

Un automate est un graphe orienté, défini par :

- un ensemble de sommets S ;
- un ensemble de relations orientées R entre les sommets S ;
- un sommet initial $x \in S$;
- un sommet final $y \in S$.

Tous les éléments de S ont au moins une relation entrante et une relation sortante, éléments de R, sauf le sommet initial et final qui ont respectivement une seule relation sortante et une seule relation entrante. Un chemin est un parcours d'automate qui commence par le sommet initial et se termine par le sommet final. Dans les représentations classiques de graphes, les sommets sont représentés par des nœuds, et les relations par des arcs.

Pour l'application en TAL, les sommets expriment des contraintes (comme par exemple des suites de caractères, une catégorie grammaticale) qui sont testées sur le texte donné en entrée. Le parcours est réussi si un chemin peut être complété.

Les grammaires de détection d'entités à Sinequa sont encodées dans des automates à états finis. Il existe deux formalismes, selon le niveau de traitement où ils sont déployés. Les premiers, les plus anciens historiquement, sont codés dans un format XML propriétaire. Pour des raisons d'articulation avec les autres traitements, un sommet correspond à une unité textuelle après projection du lexique sur le texte. Leur déploiement est totalement optimisé pour prendre en compte les informations fournies par les autres traitements linguistiques et ce sont des ressources éditées par l'équipe linguistique de Sinequa. Les seconds sont des automates qui interviennent en fin de chaîne et qui interprètent le texte comme si c'était une longue expression régulière. Ces automates peuvent être édités par les clients pour qui l'analyse proprement linguistique reste cachée.

3.4 Trois opérations dans la vie des ressources

Le cycle de vie d'une ressource est illustré dans la figure suivante. Trois étapes peuvent être distinguées : mise en place, vie et mort de la ressource. Chaque étape est caractérisée par des opérations différentes.

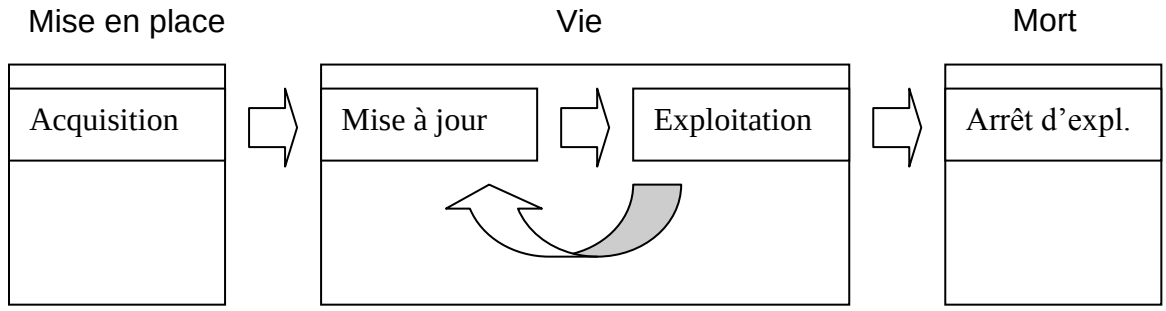


Figure 7 : Cycle de vie des ressources

L'*acquisition* et la *mise à jour* sont des opérations de modification qui se distinguent surtout par l'ampleur des modifications, grande pour l'acquisition et moindre pour la mise à jour.

3.4.1 Acquisition

L'*acquisition* correspond à la mise en place initiale de la ressource. Il s'agit d'une opération qui peut être de très grande envergure, à moins que les ressources ne soient prêtes à l'achat. Dans ce cas, on peut se contenter d'adapter le format, même si des modifications plus conceptuelles peuvent s'imposer.

Du point de vue de l'entreprise, l'acquisition de lexiques et de corpus peut se faire en interne, en partenariat avec une université, en téléchargeant des ressources sur internet, par achat chez ELDA et/ou LDC, ou par une combinaison de ces moyens-là. Seulement dans les deux premiers cas le format des ressources est naturellement proche des besoins d'entreprise et leur intégration dans le produit possible à faible coût. Les grammaires étant fortement liées à leur exploitation, nous n'en connaissons pas qui peuvent être acquises indépendamment de l'application. L'acquisition se fait, selon la ressource, de façon automatique par apprentissage ou manuelle par le codage des règles.

Même si de plus en plus de lexiques et de corpus sont disponibles auprès de distributeurs comme ELDA et LDC, l'acquisition d'une ressource peut être étonnamment pénible. Peu de ressources correspondent en général exactement au traitement qu'on veut effectuer, et le prix d'achat est souvent prohibitif. Ces deux raisons sont à l'origine du redéveloppement de ressources qui sont censées exister et être disponibles dans la communauté du TAL. Nous ne pouvons que saluer tous les efforts faits pour mettre en place des infrastructures pour ouvrir les ressources à la communauté. Certaines sont centralisatrices, comme le portail du CNRTL[25] ou du TST-Centrale[26] belgo-néerlandais, d'autres sont distribuées, comme la structure de partage de ressources international du MPI de Nimègue où chaque site met à disposition du réseau au moins les métadonnées de ses ressources.

[25] Centre National de Ressources Textuelles et Linguistiques.Voir : http://www.cnrtl.fr/.
[26] Voir : http://www.inl.nl/tst-centrale

Pendant notre thèse, nous avons mis en place, en encadrant un ou plusieurs locuteurs natifs, toutes les ressources nécessaires des langues suivantes pour les traitements faits dans la technologie de Sinequa : arabe, danois, grec, finnois, italien, néerlandais, polonais, portugais, suédois

3.4.2 Mise à jour

La mise à jour couvre quatre opérations dont le déroulement est illustré dans la figure 8. Deux cycles doivent se terminer avant qu'on procède à l'opération finale qui est celle du figement qui ouvre la voie à l'exploitation. Le premier cycle est celui de modification de la ressource suivi du test du traitement. Une modification est un ajout, une suppression ou une combinaison des deux. Tant que le test n'est pas réussi, on revient à la modification. Cette phase de test peut très bien comporter une évaluation du traitement. Quand le test réussit et les résultats de l'évaluation sont suffisants, on passe à la validation. Celle-ci consiste à exécuter un ensemble de tests préétablis qui vérifient que le traitement correspond bien à ce qu'il était censé faire et ne montre pas de régression par rapport au précédent état des ressources. La validation s'inscrit donc dans un historique, et repose sur des corpus de validation, c'est-à-dire des ensembles de documents annotés qui servent à valider le traitement visé et donc la mise à jour des ressources. Tant que la validation n'est pas totalement réussie, on revient sur le cycle modification-test. Si la validation réussit, alors on passe à l'étape de figement, où on fige l'ensemble des ressources qui forment un ensemble cohérent avec les ressources de traitement. Il est alors possible de leur attribuer un numéro de version et elles sont prêtes pour être exploitées.

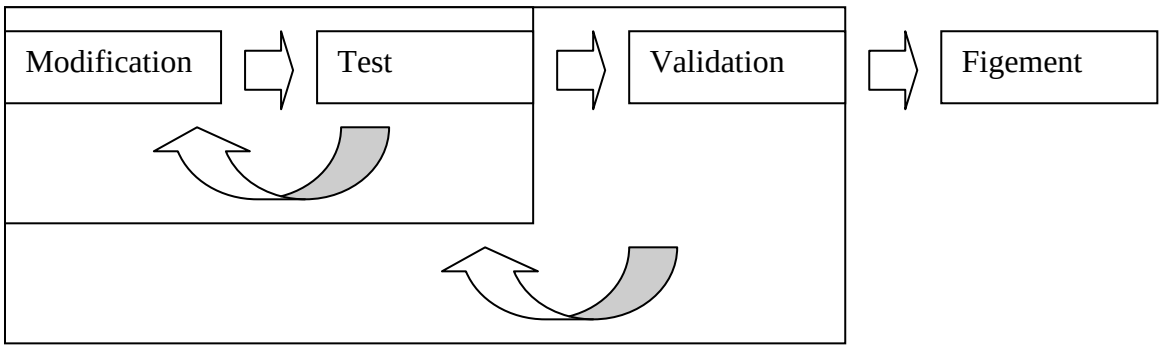


Figure 8 : Les différentes étapes de la Mise à jour

Le lecteur aura remarqué que nous testons, évaluons et validons un traitement, et non pas une ressource linguistique. La validation de la ressource est donc indirecte.

La validation d'une ressource sans passer par un traitement est un non sens à cause de la nature même d'une *ressource*. Une ressource ne peut être validée ni évaluée *in abstracto*, sauf sur des critères intrinsèques comme le respect de la syntaxe. Pourtant, tout linguiste informaticien est capable, en vue d'un certain traitement, de qualifier une ressource lexicale presque à vue d'œil. Pour ce faire, il procède à un échantillonnage rapide, vérifie la cohérence des informations tout en les confrontant à ses propres connaissances. C'est exactement de

cette façon que le centre de validation de ressources écrites[27], le Center for Sprogteknologi (CST), procède à l'évaluation rapide des ressources. La méthode du Quick Quality Check (QQC) décrite dans [Fersøe et Olsen, 2005] correspond à une version simplifiée d'une procédure de validation très détaillée ([Fersøe, 2004]). Elle se concentre sur trois aspects : la documentation, le format et le contenu. La simplification est motivée par une réduction du temps nécessaire à la validation : remplir une QQC ne devrait prendre que 5 à 6 heures. La validation du contenu reste très limitée : le codage d'un échantillon de 30 mots seulement est regardé. Dans le cadre d'une évaluation profonde, [Fersøe, 2004] donne des indications très intéressantes sur l'échantillonnage dans des lexiques de grande couverture : il faut examiner 100% des mots des classes fermées, et ensuite un échantillon représentant 1 à 2 % des entrées des classes ouvertes avec un minimum de 1000 mots pour chaque classe. Néanmoins, même après cette validation, nous ne savons toujours pas ce que vaut la ressource pour un certain traitement tant qu'elle n'a pas été intégrée.

Evaluer une ressource dans un traitement veut aussi dire la comparer avec une autre version de la même ressource ou bien avec une autre ressource. Pour comparer deux ressources, il faut les mettre dans la même situation d'exploitation, et mesurer la performance du traitement dans les deux cas. Cela permettra ensuite de conclure que dans le cadre d'évaluation choisi, l'une des deux ressources est plus adaptée que l'autre pour effectuer le traitement considéré. Dans la plupart des cas, une telle évaluation est très lourde à mettre en œuvre. L'intégration de ressources peut demander des moyens considérables, à cause de différences qui peuvent aussi bien être liées au format que de nature conceptuelle.

Le modèle de langage qui sert à la désambiguïsation morphosyntaxique est obtenu à partir d'un entraînement et est de ce fait le plus difficile à mettre à jour. En effet, l'entraînement peaufine le modèle jusqu'à trouver la grammaire et le lexique les plus efficaces sur le corpus d'entraînement. L'état de l'art de l'étiquetage morphosyntaxique mentionne des performances de 95 à 98 % pour certains types de textes, ce qui laisse quand même une bonne marge d'erreur : si la phrase moyenne était de 20 mots, 5% d'erreur voudrait dire que toute phrase contient une erreur. Ces modèles n'étant pas parfaits, on peut être tenté de mettre ces modèles à jour en reprenant le lexique ou la grammaire à la main. Par l'ajout d'une règle ou une modification dans le lexique, on met cependant en danger l'équilibre et l'efficacité du modèle. Il est donc déconseillé de le faire par une mise à jour, mais d'agir sur le corpus d'entraînement, en ajoutant ou supprimant des informations, et de relancer l'entraînement pour obtenir un nouveau modèle. Cette procédure est bien plus complexe, mais garantit la cohérence du modèle.

Durant notre thèse, nous avons mis à jour ou encadré la mise à jour de la quasi-totalité des dix-neuf langues au catalogue de Sinequa, avec un travail plus important sur les langues commercialement stratégiques. Pour donner une idée du nombre de changements, nous avons chiffré le nombre de modifications faites dans cinq langues sur une période de sept semestres successifs. Ces chiffres de mise à jour des lexiques morphosyntaxiques sont donnés dans le chapitre 8.3.

Avec l'équipe linguistique, nous avons mis en place les processus de validation pour chacune de ces langues. Celles-ci reposent sur des corpus de validation qui sont composés de corpus annotés décrivant le comportement voulu d'un traitement. Comme ces corpus doivent décrire

[27] Il existe pour le moment deux centres de validation de ressources pilotés par VCom, le comité de validation d'ELRA. Le SPEX, au Pays-Bas, valide les ressources de l'oral, le CST au Danemark valide les ressources écrites. Les rapports de ce dernier sont consultables sur son site web : http://cst.dk/validation.

l'ensemble des règles contenues dans le traitement, ils sont mis à jour au fur et à mesure que les traitements évoluent.

3.4.3 Exploitation

Dans la phase d'exploitation, ressources de traitement et ressources linguistiques ne font plus qu'un. Dans cette phase on n'intervient pas sur les ressources, à moins de retomber dans le cycle de la mise à jour. Une évaluation sur des bases scientifiques ou des tests industriels du traitement a lieu dans les conditions d'exploitation.

C'est aussi l'exploitation qui conditionne l'adaptation des ressources lors des phases précédentes que sont l'acquisition et la mise à jour. Nous distinguons l'adaptation interne et externe des ressources.

Nous appelons *adaptation interne* le fait d'adapter les traitements et les ressources pour prendre en compte les particularités ou les erreurs des traitements précédents. Dès les premiers traitements, il existe des particularités ou des erreurs dont il faut tenir compte dans les niveaux supérieurs. Les ressources sont alors adaptées pour les prendre en compte. Nous pouvons donner comme exemple d'erreur une mauvaise extraction du texte d'un document PDF ou une fausse identification de la langue sur l'espace de quelques mots. Une particularité serait par exemple l'absence de règle pour un nom adjectival dans le modèle de désambiguïsation alors que l'étiquette morphosyntaxique existe dans les lexiques, provoquant des étiquetages inattendus.

Nous appelons *adaptation externe* le fait que les traitements et les ressources sont spécifiques à l'application, indépendamment de l'adaptation interne. Par exemple la désambiguïsation morphosyntaxique dans un système de RI peut être explicitement conçu pour laisser l'ambiguïté sauf si le système est sûr de ne pas se tromper, pour éviter de provoquer du silence.

La combinaison de l'adaptation externe et interne fait que les traitements et ressources sont adaptés au système dans lequel ils sont déployés.

3.5 Des outils aux environnements de gestion

Pour les personnes qui ne sont que faiblement exposées aux ressources, la *gestion* de celles-ci reste souvent une notion assez floue, au point d'être confondue avec des normes, des logiciels, des plateformes et des théories. Tout cela est bien sûr lié au sujet des ressources, mais pas forcément à leur gestion. Cette confusion est assez naturelle, car de nombreux outils intègrent un minimum d'interfaçage de gestion dans les limites du traitement qu'ils proposent. La fonctionnalité proposée est généralement liée à la mise à jour des données lexicales ou l'édition de règles. Si cette interface est indispensable, il manque d'autres fonctionnalités pour répondre à toutes les besoins en gestion. Avant d'exprimer ces besoins, nous passons en revue les outils et les systèmes qui sont souvent confondus avec un environnement de gestion et ceux qui les fournissent plus ou moins.

3.5.1 A ne pas confondre avec…

Des plateformes d'annotation[28]

Les laboratoires spécialisés dans le TAL ont souvent développé des plateformes d'annotation pour automatiser et rendre générique l'enchaînement des différents traitements en composants. L'idée est de limiter le travail nécessaire pour enchaîner les traitements et de pouvoir modifier la chaîne sans avoir à recoder toutes les interfaces entre les traitements. Comme les modules s'interfacent tous avec le cadre commun, ils intègrent tous la même norme d'entrée/sortie. Les modules sont donc échangeables et réutilisables, peu importe le langage de programmation

GATE, *General Architecture for Text Engineering* ([Cunningham et al., 2002]), se décrit comme une boîte à outils pour le *text mining* développé à l'université de Sheffield. Tout comme pour LinguaStream ([Widlöcher et Bilhaut, 2005]) de l'université de Caen, ou encore Ellogon ([Petasis et al., 2002]), il est possible pour l'utilisateur de modifier à sa guise les chaînes de traitement par le biais d'une interface visuelle. Ces trois plateformes contiennent des outils tous faits pour le TAL comme par exemple des modules d'édition de ressources (un éditeur de lexique sémantique par exemple) ou d'extraction d'entités nommées (avec un éditeur de graphe) ou de visualisation de statistiques. Les auteurs de [Heinecke et al., 2008] présentent TiLT, la plateforme extrêmement modulaire d'Orange Labs.

En proposant une normalisation des entrées et sorties, UIMA, *Unstructured Information Management Architecture* ([Ferrucci et Lally, 2004] et [Ferrucci et al., 2006]), rend possible l'enchaïnement de modules totalement indépendants. L'industrie semble bien apprécier l'arrivée de cette norme sous licence libre développée dans un « esprit industriel ». UIMA a en effet été développé à l'origine par IBM pour répondre à des besoins d'interopérabilité de traitements développés en interne. Les composants d'une plateforme respectant UIMA sont ainsi plus facilement remplaçables par d'autres composants respectant la même norme.

La valeur ajoutée de ces plateformes est la possibilité d'échanger librement les traitements. Or, à Sinequa, nous n'avons pas ce besoin : l'ordre d'exécution des modules est défini, et la notion de module n'est pas toujours respectée pour des raisons d'optimisation du code. Les autres fonctionnalités, notamment la visualisation de l'annotation, le traitement des erreurs et le pilotage d'une série de traitements sont plus intéressantes pour nous car elles s'approchent de la gestion des ressources.

Des boîtes à outils pour les traitements[29]

NLTK, *Natural Language Toolkit*, est un ensemble d'outils en python avec une grande valeur pédagogique car il est accompagné d'un livre téléchargeable pour s'initier au TAL.

Des environnements de programmation

Les grammaires peuvent être vues comme des langages de programmation. Pour la plupart des langages de programmation, il existe des IDE (*Integrated Development Environment*) qui facilitent la vie au programmeur en offrant à la fois les fonctionnalités d'édition, de compilation, d'exécution et de déboguage. *Eclipse* est un tel environnement de programmation initialement créé pour JAVA.

---

[28] GATE : http://gate.ac.uk ; LinguaStream : http://www.linguastream.org ; UIMA : http://incubator.apache.org/uima

[29] NLTK : http://www.nltk.org/

Avec le développement de grammaires plus complexes, c'est-à-dire des automates à états finis, l'équipe de Sinequa a mis en place un IDE qui est plus qu'un simple éditeur d'automates. Depuis l'interface il est notamment possible d'exécuter l'automate édité sur des fichiers prédéfinis (validation) et sur des phrases, ainsi que d'exécuter des scripts. L'interface vérifie également les catégories grammaticales alors que les contraintes lexicales de type mot-forme et lemme ne sont pas testées, le lien direct avec les lexiques étant inexistant.

Des éditeurs d'ontologies

Une ontologie est un modèle des connaissances d'un domaine donné. Les entrées sont des concepts liés par des relations sémantiques dans le but de pouvoir raisonner dessus. Il existe des éditeurs d'ontologie permettant de créer la structure, la remplir et la mettre à jour par une interface graphique. Ils contiennent en général aussi un moteur de raisonnement qui permet de raisonner sur les connaissances de l'ontologie. Le plus connu parmi les éditeurs d'ontologie est sans doute Protégé de l'Université de Stanford, né de la recherche en ingénierie linguistique biomédicale. Il en existe d'autres comme KAON, DOE, HOZO[30].

Nous avons essayé de détourner l'interface de Protégé pour coder un thésaurus multilingue des pays du monde avec leurs régions et leurs villes en OWL. Ce fut un échec à cause de l'impossibilité de créer des liens multilingues de façon efficace. Autre inconvénient, l'éditeur graphique demande un grand nombre de clics rendant inefficace l'interface pour l'encodage de lexique. La structure logique d'une ontologie n'est pas faite à l'origine pour contenir des informations lexicales, mais sémantiques. Cet échec n'est donc pas une surprise complète.

Des normes ou des recommandations d'encodage

La TEI contient des recommandations pour l'encodage de corpus et d'entrées lexicographiques de dictionnaires traditionnels. Pour l'encodage des lexiques, Sinequa pourrait s'inspirer des résultats de Genelex ou de la norme ISO 24613:2008, mais comme il n'y a pas de partage de ressources et qu'il n'existe pas d'outils prêts à l'emploi, l'intérêt reste très limité.

Des théories lexicales ou linguistiques

Les choix faits pour l'encodage sont motivés par l'efficacité des traitements. Les théories linguistiques ou lexicales comme les grammaires d'unification dans le sillon de LFG ([Kaplan, 1982]), le lexique grammaire de Gross ([Gross, 1968]) dans le courant des grammaires transformationnelles ([Harris, 1964]), ou le sens-texte de Mel'çuk ([Mel'čuk, 1973]), ne sont pas prises en considération, aussi intéressantes soient-elles. Les traitements ne font pas de présupposition théorique et n'essaient pas d'apporter de preuve applicative à une théorie linguistique. Leur but est de rendre possible les traitements dans la limite des spécifications définies.

Des langages de codage lexical

DATR est un langage pour la description lexicale qui est indépendant d'un cadre linguistique particulier. Il utilise le principe de l'héritage et se prête donc bien à une description de lexique hiérarchique, typiquement les lexiques utilisés dans les grammaires d'unification. Ce format est très intéressant pour tester rapidement des théories, mais demande un outillage spécifique à l'exploitation.

---

[30] Protege : http://protege.stanford.edu/ ; KAON : http://kaon.semanticweb.org/ ;
DOE : http://homepages.cwi.nl/~troncy/DOE/ ; HOZO : http://www.ei.sanken.osaka-u.ac.jp/hozo/

Des concordanciers

The Sketch Engine [Kilgarriff et al., 2004] est un concordancier payant en ligne. Ces fonctionnalités avancées et la disponibilité des grandes langues européennes en font un outil par excellence pour tout travail lexical mono- et multilingue. Ce genre d'outil est un instrument de travail indispensable pour tout travail lexicologique mais n'est pas un outil de gestion en soi.

3.5.2 Outils et plateformes de gestion existantes

L'université de Grenoble a une longue expérience en édition de dictionnaires pour la traduction automatique, aboutissant aujourd'hui à une proposition pour un « méta-Environnement de Développement Linguiciel » pour la TAO[31] comme l'appelle [Nguyen et Boitet, 2007].

A la même université, l'une des applications apparues au cours de ces travaux est le dictionnaire multilingue résultant du projet Papillon[32], détaillé dans [Mangeot et al., 2003] et issu des travaux de thèse de [Sérasset, 1994]. Gilles Sérasset proposait avec SUBLIM (Système Universel de Bases Lexicales Multilingues) un système d'édition lexicale dans lequel le linguiste peut définir lui-même la structure de données et faire un mixte de différentes structures de données au sein d'une même base de connaissances. [Sérasset, 1994] décrit 3 niveaux dans son architecture : accès (visualisation), interne (manipulation) et base de données (stockage) et propose 5 outils : éditeur, navigateur, vérificateur de cohérence, défauteur, import/export. Il ajoute que la situation idéale est de fusionner navigation et édition, puisque c'est en naviguant qu'apparaît souvent le besoin d'éditer.

Ce travail a été continué par Mathieu Mangeot avec le développement de Jibiki, une « plateforme de gestion de ressources lexicales », utilisée dans les projets Papillon[33], GDEF[34] et LexAlp[35]. Papillon est une base lexicale multilingue à structure pivot couvrant des langues européennes et asiatiques. La structure des entrées est construite selon les principes du DEC[36]. Le but du projet GDEF est de réaliser un dictionnaire estonien-français de 80.000 articles. Jibiki est aussi utilisé dans le projet LexALP, qui vise l'harmonisation de la terminologie juridique transalpine en quatre langues : allemand, français, italien et slovène. Cette plateforme est disponible pour d'autres projets à condition que « les ressources produites avec la plate-forme [soient] disponibles au public et libres de droits ». Jibiki est décrit dans [Chalvin et Mangeot, 2006] comme « un environnement générique en ligne permettant la rédaction et la consultation de tous types de dictionnaires : glossaires terminologiques, dictionnaires bilingues, bases lexicales multilingues, etc. ». Cette description montre que cette plateforme est à la fois trop riche et trop pauvre pour correspondre à nos besoins. Les dictionnaires dont nous disposons sont des lexiques, qui ne sont pas rédigés, loin de là, et il n'existe pas de cycle de rédaction. Nos données sont majoritairement monolingues, même si la structure à pivot semble très intéressante pour nos travaux sur le multilinguisme du moteur. En l'absence de nombreux utilisateurs répartis dans le monde, nous n'avons pas besoin d'une architecture web évoluée ; au contraire, cela nous semble plutôt dangereux pour la sécurité

[31] Traduction Assistée par Ordinateur (TAO).
[32] Le dictionnaire et la documentation sont disponible sur http://www.papillon-dictionary.org/.
[33] Site web du projet : http://www.papillon-dictionary.org/
[34] Grand Dictionnaire Estonien Français (GDEF). Pour plus d'informations, voir http://estfra.ee/
[35] Système d'harmonisation de la terminologie juridique sur l'environnement et l'aménagement du territoire dans les Alpes multilingues (LexAlp), janvier 2005-2008. Voir http://lexalp.free.fr/ pour une description complète.
[36] Dictionnaire Explicatif et Combinatoire (DEC), [Mel'čuk et al., 1995].

des données. Les idées qui ont mené à la mise en place de la plateforme sont néanmoins très intéressantes : les besoins de modélisation des données de l'utilisateur (= le linguiste) priment sur le codage même des données.

[Joffe et de Schryver, 2004] présentent l'éditeur commercial TshwaneLex pour l'édition de dictionnaires électroniques mono- ou bilingues. C'est un outil lexicographique qui part du principe que le lexicographe n'est pas un spécialiste de l'informatique et fournit donc toutes les fonctions nécessaires pour construire des dictionnaires. L'outil s'interface nativement avec des bases de données. Il prévoit l'importation de données, des recherches et des filtres, des tests d'erreurs de formatage, des fonctionnalités d'édition spécifiques au bilinguisme, la gestion automatique des références. Il gère également le travail en équipe en interdisant l'accès à des entrées qui sont en cours de modification.

[Courtois, 1995] décrit les différents types de lexiques qui peuplent l'univers DELA utilisés dans Intex et Unitex : DELAS et DELAF pour les mots simples, DELAC et DELACF pour les mots composés, DELAP et DELAPF pour les lexiques phonémiques, tables syntaxiques et automates. DELAF et DELACF sont des lexiques de formes générés, avec des opérations plus ou moins simples, à partir respectivement du DELAS et du DELAC. En effet, les entrées des DELAS et DELAC sont des lemmes accompagnés d'au moins une classe de flexion exploitée à cette fin. S'y ajoute le lexique syntaxique, appelé lexique-grammaire.

NooJ [Silberztein, 2003] est la réécriture d'Intex en C#. Comme nous pouvons lire dans [Silberztein, 2005], la gestion de cinq différents types de lexiques (mots simples, mots composés, et les tables syntaxiques du lexique-grammaire) a été totalement simplifiée : il n'existe plus qu'un seul lexique. L'interface de NooJ permet entre autres l'importation d'un corpus et le calcul de quelques statistiques simples sur ce corpus comme la fréquence des mots. Elle indique également les mots inconnus des lexiques, fonction importante pour détecter les trous lexicaux. NooJ contient en outre un concordancier, permettant d'étudier l'emploi d'un mot dans son contexte gauche et droit. L'éditeur de grammaire est sans doute l'interface principale où l'on édite les automates qui vont s'appliquer sur le corpus. Elle permet de rattacher à l'automate des exemples et contre-exemples qui servent de test pour l'automate. Une autre fenêtre permet de regarder en détail l'annotation morphosyntaxique.

Aleth est l'ensemble de ressources et d'outils commercialisé par feu GSI-ERLI. Parmi les applications on compte la recherche documentaire, la génération de texte, la traduction etc. Tous les traitements reposent sur la même ressource : une base de connaissance linguistique au format Genelex. Depuis le projet Genelex, toutes les connaissances lexicales ont été centralisées, et des lexiques spécialisés sont extraits en fonction des projets (d'après la description dans [LIM, 1994]). AlethGD, aussi connu comme *Station Genelex*, est le système de gestion de cette base de connaissances qui assure « (1) la consultation et la visualisation de l'ensemble des données, (2) la mise à jour (création, suppression et modification des données du dictionnaire) et (3) les opérations massives sur le dictionnaire : suppression et fusion de listes de données extérieures dans le dictionnaire. Station Genelex permet en outre de réaliser (4) des opérations de contrôle de cohérence sur le fond comme la détection des unités morphologiques qui n'ont pas de flexion ou encore la détection des singularités de codage morphologique pour les composés. Enfin, Station Genelex offre différents services, (6) un générateur morphologique de mot simple ainsi que (7) des opérations de tri sur le dictionnaire. » ([UHB, 2001][37]) Pour les opérations d'import et d'export les linguistes disposaient entre autres d'un langage de programmation lexicale orientée objet ([Humphreys, 1996]).

[37] Citant [Francopoulo et al., 1992] que nous n'avons pas pu consulter.

Le laboratoire LPL de l'Université de Provence a développé une plateforme de développement lexical pour la maintenance du lexique DicoLPL. Ce lexique est codé en extension au format texte tabulaire : chaque ligne est un quadruplet <mot, phonétisation, catégorie, lemme>. Si le titre de [VanRullen et al., 2005] promet des fonctionnalités d'acquisition, de maintenance et de validation, l'accent est surtout mis sur l'enrichissement du lexique. Ils présentent un outil de fréquençage qui permet également de lister les mots inconnus du lexique. Les informations fréquentielles du lexique sont destinées à servir de filtre pour définir un lexique noyau pour des traitements sémantiques.

3.5.3 Du besoin d'un environnement de gestion de ressources linguistiques

Le contexte dans lequel nous nous trouvons est celui d'une plateforme d'analyse linguistique, où plusieurs traitements font appel à des informations qui sont codées dans plusieurs ressources. Les traitements et leurs ressources sont organisés de façon *analytique* : à chaque traitement correspondent des ressources qu'on ne mélange pas entre elles. Il en découle que les formats des ressources peuvent être d'une grande simplicité, et donc facilement manipulables. En l'occurrence, les lexiques sont codés dans un format texte avec une syntaxe particulière qui varie selon la ressource. Le linguiste est donc totalement autonome avec les ressources et ne dépend d'aucun logiciel pour les manipuler. L'autre avantage est qu'il est alors facile d'ajouter ou d'enlever un traitement ou une ressource, comme par exemple ajouter un lexique spécialisé pour un client en particulier. Les redondances dans les informations entre les ressources sont le principal inconvénient.

Dans une organisation *synthétique*, toutes les connaissances sont contenues dans la même ressource, mais cela demande une infrastructure informatique de gestion complexe et donc un investissement important dans la maintenance de cette infrastructure. Dans ce cas, la place du linguiste est plutôt confortable car les tâches de gestion (versioning, sauvegarde, livraison, etc.) sont prises en charge par l'infrastructure et il peut se concentrer sur l'analyse linguistique. Mais comme l'évolution de cette infrastructure dépend de l'informatique, le linguiste peut se trouver bloqué le temps de la faire évoluer. Cette dépendance ne plaide pas en faveur de ce type d'organisation, même si elle est conceptuellement la plus simple.

Notre approche se situe entre les deux : ne pouvant toucher à l'architecture actuelle nous respectons la vision analytique des ressources, mais nous simulons la vision synthétique des ressources par la mise en place notamment d'un accès unique à toutes les informations lexicales.

Nous cherchons à mettre en place un *environnement* de gestion de ressources linguistiques plutôt que d'un système ou d'une plateforme. Ces deux derniers termes supposent en effet une certaine unité logicielle que nous ne pouvons pas assurer. L'ensemble des outils nécessaires que nous avons mis en place sont complémentaires, peuvent être déployés en même temps mais ne sont pas toujours liés entre eux, ce qui est un minimum pour parler d'une plate-forme. Une plateforme est surtout une structure d'accueil qui permet de combiner des outils à volonté.

L'environnement de gestion idéal se charge de toutes les opérations de gestion nécessaires sur les ressources, c'est-à-dire de l'acquisition et du cycle de la mise à jour qui comprend la validation. Il faut donc combiner les fonctionnalités qu'on trouve dans les outils existants :

- un éditeur de lexique intégré avec un accès unique non intrusif, c'est-à-dire qui donne une vue intégrale sur l'ensemble des informations lexicales et qui n'impose pas son utilisation pour des modifications massives qui sont plus faciles à faire par script ;

- un éditeur de grammaires qui fait le lien avec l'éditeur de lexique ;
- une visualisation de l'annotation paramétrable qui distingue les différents niveaux d'annotation pour ne pas être noyé dans les annotations ;
- des outils d'étude linguistique pour étudier les faits qui sont à l'origine des règles et du lexique, comme par exemple un concordancier intelligent ;
- un outil de mise à jour du lexique qui facilite la surveillance de l'évolution de la langue, particulièrement des néologismes, et aussi l'ajout dans le lexique.

Les lexiques et grammaires que l'équipe de Sinequa manipule sont tous au format texte, ce qui a le grand avantage de faciliter l'accès à ces fichiers et leur manipulation par des langages de script. Cette situation est un choix stratégique fait par la direction de Sinequa pour qui l'autonomie totale du linguiste est primordiale. Il y a donc deux exigences contradictoires : d'un côté que le linguiste soit totalement autonome, et de l'autre que toutes ses opérations soient codifiées par des interfaces avec une structure très contraignante derrière. Même si cela semble impossible, nous allons essayer de concilier les deux et de garder le meilleur des deux mondes.

Notre expérience fait néanmoins apparaître que la méthode d'acquisition a été assez variable d'une langue à l'autre selon les ressources qui sont intégrées. Cela rend difficile la mise en place d'un environnement unique. Il est bien sûr possible d'imaginer des outils d'augmentation lexicale massive et des aides pour accélérer la mise en place des grammaires, mais l'intégration de cet outillage n'est nécessaire que si l'accès à la base de connaissances n'est pas évident. Dans notre cas, ce n'est pas une base de données, mais un ensemble de fichiers texte dont l'accès est particulièrement facile, ce qui n'exclut pas une connexion entre les outils.

Pour la mise à jour lexicale, le principal outil à disposition est un fléchisseur. Pour la plupart des langues, des classes et des règles de flexion ont été développées pour générer à partir d'un lemme et de sa classe de flexion toutes les formes et leurs descriptions à ajouter dans le lexique. Là encore, cet outil ne doit pas forcément être intégré dans une interface de saisie lexicale tant que les lexiques sont codés en fichiers texte.

3.6 Conclusion

Les ressources linguistiques fournissent les informations aux traitements mis en œuvre dans une application. Pour assurer la qualité de cette exploitation, il est important de veiller à la cohérence des informations linguistiques pendant les opérations de gestion : dès l'acquisition et lors de leur mise à jour. Les lexiques qui sont activement gérés – ou qui l'ont été – ont en général un outil de codage qui tient compte des spécificités de leur format. Or, les besoins vont plus loin dans un environnement applicatif analytique où il existe des ressources pour chaque traitement. Les informations linguistiques se trouvent ainsi dispersées sur plusieurs ressources, avec un risque d'incohérence accru. Lexiques, corpus et grammaires doivent être concordants à chaque moment de leur cycle de vie. Cela demande la mise en place d'un environnement de gestion de ressources linguistiques.

Chapitre 4

Lexiques et formalisation lexicale

L'émergence du terme *architecture linguistique* est intimement liée à l'histoire du TAL, et plus particulièrement au développement des lexiques pour le TAL. Les lexiques sont souvent au cœur des techniques mises en œuvre. Comme la constitution de ces lexiques est un travail de longue durée et demande un investissement conséquent, le partage de ces ressources a été très rapidement vu comme un enjeu de taille. C'est dans cet esprit que des organisations comme ELRA ont vu le jour[38]. Il ne suffit néanmoins pas de mettre ces ressources à disposition de tout le monde, il faut encore que le formatage des données soit suffisamment généraliste pour qu'elles puissent convenir à tous, d'où la volonté de normaliser les formats. Le terme d'architecture linguistique est apparu au cours de ces travaux de normalisation. C'est au croisement de la lexicologie et du TAL que les recherches aboutissent à la définition de structures plus générales pour l'encodage d'informations linguistiques. Dans ce chapitre nous suivons le chemin parcouru à travers ces projets et examinons quelques structures lexicales proposées dans la littérature avant de définir ce que couvre le terme d'*architecture linguistique*.

[38] Comme on peut lire dans [Mariani, 1995], rapport du projet SPEECHDAT, un des projets européens à l'origine de la fondation d'ELRA.

4.1 Le long chemin de généralisation des structures lexicales

Tout lexique a une structure lexicale, même si elle n'est souvent pas explicitée. Le souhait de partager les ressources a mené à des travaux de généralisation de ces structures lexicales afin de diminuer les coûts d'adaptation. Les différentes normes, recommandations et directives ont toutes une longue histoire.

4.1.1 La tentation d'exploiter l'existant

Si l'avènement de SGML au milieu des années 80 ouvre la voie au partage et à l'archivage à long terme de documents textuels dans un monde dominé par des systèmes documentaires aux formats propriétaires, il a aussi des répercussions sur l'édition des dictionnaires. Afin d'être exploités par les systèmes d'impression, les dictionnaires étaient auparavant encodés en texte entrecoupé par des commandes typographiques, et les éditeurs vont progressivement procéder à la conversion de leurs données.

La communauté du TAL s'est intéressée très tôt à ces grands dictionnaires dont le contenu est le fruit de longues années d'édition par des lexicographes expérimentés. Comme on peut lire dans [Evens et Smith, 1979], la conception d'un lexique est entravée d'un grand nombre de considérations théoriques et philosophiques dont la résolution influe directement sur le codage des données. Le volume des données est tel que tout choix de codage peut être irréversible.

Comme la constitution de lexique est un travail de longue haleine, qu'à cette époque-là, les ordinateurs n'avaient pas encore leur puissance actuelle et que les grands corpus n'existaient pas encore, beaucoup de chercheurs se sont lancés dans la recherche pour récupérer les connaissances contenues dans les grands dictionnaires papier. Ils espéraient notamment exploiter les définitions pour créer des bases de connaissances sémantiques complètes. La thèse de [Amsler, 1980] peut être mentionnée ici comme l'un des travaux précurseurs dans ce domaine.

Environ 15 ans plus tard, les chercheurs en TAL concluent à l'échec de l'exploitation automatique de ces connaissances. [Atkins, 1991] appelle *human bias* le fait que les dictionnaires sont faits pour des lecteurs humains, et non pas pour une utilisation computationnelle, et qu'ils peuvent de ce fait comporter des incohérences qui ne heurtent pas les lecteurs. En confrontant à la réalité du corpus des entrées tirées de cinq dictionnaires il montre que les dictionnaires peuvent contenir des descriptions sémantiques contradictoires. En conclusion, le titre de [Ide et Véronis, 1993][39] en dit long : *Extracting knowledge bases from machine-readable dictionaries: Have we wasted our time?* En dépit de quelques travaux comme la construction de Mindnet par [Richardson et al., 1998], il fallait attendre l'apparition de Wikipedia pour que la communauté se repassionne pour l'exploitation des connaissances dictionnairiques.

Si l'exploitation de dictionnaires électroniques[40] par les chercheurs en TAL n'a pas été concluante, ces recherches ont tout de même rapproché la communauté lexicographique de

[39] Republié en [Ide et Véronis, 1995a]

[40] Dictionnaires électroniques : traduction de *machine readable dictionnaries*, qui désigne la version électronique d'un dictionnaire papier.

celle du TAL[41]. L'approche plus formalisée de cette dernière mènera à des tentatives de normalisation de l'encodage des dictionnaires dans la technologie de l'époque : SGML. [Amsler, 1980] initie le mouvement avec le constat qu'un grand nombre de chercheurs travaillent sur les mêmes données sans mutualiser les efforts de transcodage et de correction des données. Cette prise de conscience aboutira au début des années 90 à l'inclusion du chapitre 12 *Print Dictionaries*, dictionnaires papier, concernant le formatage de dictionnaires dans la TEI/P3[42]. [Ide et Véronis, 1995b] décrivent les principaux problèmes rencontrés dans la définition de cette partie des recommandations.

La *Text Encoding Initiative* (TEI) est née d'une réunion internationale à Vassar College en novembre 1987 et est créée officiellement l'année d'après. Après des années de réunions d'experts venant de tous bords et deux ébauches (P1 en 1990 et P2 en 1992), les recommandations sont publiées en mai 1994 sous le nom de *Guidelines for Electronic Text Encoding and Interchange*, recommandations pour le codage et l'échange des textes informatisés, connu sous le nom TEI/P3 ([Sperberg-McQueen et Burnard, 1994]). Elles s'appliquent sur un grand éventail de textes différents. Les textes cibles de la réunion à Vassar College étaient « tout texte destiné au savoir humain » ([Ide et Sperberg-McQueen, 1995]). L'inclusion des dictionnaires dans la définition de texte était discutable en raison de l'absence de texte continu, mais la réunion a décidé d'y inclure ce cas limite. La TEI pourra donc être utilisée pour standardiser l'encodage de lexiques pour le TAL. [Kilgarriff, 1999] et [Erjavec et al., 2000] illustrent la difficulté d'un tel encodage : ils décrivent l'adaptation nécessaire pour faire entrer 6 dictionnaires papier dans la TEI et ensuite créer une seule DTD.

Les recommandations de la TEI vont ensuite évoluer : la TEI/P3 avait adopté SGML comme norme d'encodage, mais le succès de XML mène à une révision du codage qui se concrétise dans la TEI/P4 ([Burnard, 2001]), publiée en 2002. La dernière version est la TEI/P5 qui est sortie en novembre 2007. Elle apporte des changements sur le contenu, comme par exemple sur l'annotation des entités nommées (noms de personnes, de dates et de lieux) et l'annotation de corpus.

Comme nous pouvons lire dans [Ide et al., 1992], la TEI s'adresse à quatre types d'utilisateurs : éditeurs et lexicographes, linguistes informaticiens[43], philologues et historiens, et utilisateurs de dictionnaires. D'après ces auteurs, le but du linguiste informaticien est *typiquement* de représenter les données du dictionnaire dans une base de données lexicale, et de l'enrichir avec d'autres informations linguistiques ; la TEI leur permettrait d'échanger des informations et faciliterait la fusion. La TEI ne s'adresse donc pas à eux pour définir le format de lexiques à usage informatique, c'est-à-dire qui seront directement exploités dans des applications informatiques. Elle a été créée à partir d'observations sur la structure de dictionnaires électroniques d'un pendant papier. Le fait qu'il ne soit pas non plus prévu de coder des ambiguïtés lexicales dans un corpus annoté illustre le même point de vue : ce sont des recommandations documentaires et non pas d'échange.

La DTD de la TEI est complexe à cause de sa généricité. Pour y pallier, il existe le service PizzaChef sur le site de la TEI[44] : il permet de sélectionner les éléments qu'on veut avoir à sa

---

[41] L'intérêt est réciproque : de leur coté, les maisons d'édition s'intéressent à l'utilisation des techniques venant du TAL pour enrichir et systématiser le contenu de leurs dictionnaires, comme nous avons pu le constater à une démonstration donnée par Steve Crowdy (de Pearson Longman) le 21 juin 2005 à Barcelone pendant une réunion du projet Lirics.

[42] Pour plus d'informations sur l'histoire du TEI, voir [Ide et Sperberg-McQueen, 1995], ainsi que le site de la TEI : http://www.tei-c.org/About/history.xml.

[43] Traduction de *computational linguists.*

[44] http://www.tei-c.org/pizza.html

disposition. La TEI est en plus adaptable à ces propres besoins. [Tutin et Véronis, 1998] critiquent qu'on doit restreindre le pouvoir d'expression de la DTD dans tous les cas, car elle accepte la présence de structures qui n'existent dans aucun dictionnaire et cela même en prenant en compte que la partie dédiée aux dictionnaires. L'optionalité est une façon de rendre générique un modèle, mais ne contribue pas à la simplicité. Pour être appliqué il vaut mieux un cadre fixe limité qui reste stable dans toutes les circonstances.

4.1.2 L'époque des projets européens de grande envergure

Fin des années 80 et début des années 90, les grands projets dédiés à la recherche d'une architecture générique de lexique se succèdent[45] ou se superposent. Ce n'est pas une coïncidence que cela arrive dans cette période-là : les premiers grands corpus sont constitués et en partie annotés, et constituent des données que la recherche en étiquetage morphosyntaxique ne tardera pas à exploiter. Cet étiquetage automatique est considéré comme l'une des étapes fondamentale pour toute application en TAL. Il repose souvent sur des lexiques de grande taille, même si contradictoirement pour quelques étiqueteurs parmi les plus connus comme [Brill, 1992], le lexique appris sur le corpus étiqueté suffit. Cette époque sera aussi l'apogée de quelques projets de grande envergure de constitution de corpus, de lexiques et de création des outils pour les exploiter. Quelques-uns de ces projets avaient spécifiquement comme but le développement de lexiques de grande taille et ont consacré une partie de leurs travaux à la validation des modèles de données déjà proposés ou bien à la définition du leur.

Le projet européen Genelex (octobre 1989 – 1994), « *GENeric LEXicon* », et le modèle qu'il propose sont présentés dans [Antoni-Lay et al., 1994]. Il a rassemblé neuf partenaires industriels et académiques, tous spécialistes dans le lexique pour des applications de TAL, pour définir un modèle d'encodage qui convenait à l'ensemble des partenaires et développer des outils pour la gestion des lexiques. Le modèle est exprimé dans une DTD en SGML, et l'implémentation se fait dans le même langage. Le modèle se veut générique dans le sens où il est neutre par rapport aux théories (on peut coder les informations selon le cadre théorique choisi) et il convient pour le codage des lexiques, aux formats très divers, des partenaires. Dans sa suite, le projet européen CEGLEX[46] (mars 1995 – mars 1996) étendra le modèle sur trois langues de l'Europe centrale, en coopération avec EAGLES (voir plus bas).

Le projet Multilex (décembre 1990 – novembre 1993), *A Multi-Functional Standardised Lexicon for European Community Languages*, était en contact avec les projets européens Genelex et EUROLANG[47]. Dans le rapport final du projet, [Paprotté et Schumacher, 1993] font une proposition de norme internationale, appelé MLEXd, décrit en BNF[48]. Il s'agit d'un modèle de données lexical multilingue orienté vers les grammaires d'unification et implémenté en structures de traits. Il privilégie la description en intension par le lemme, reposant sur l'héritage et les règles lexicales. Les outils qui ont été développés pour la gestion

---

[45] Nous n'entrons pas dans le détail des programmes qui ont financé les projets. La logique de financement de certains de ces projets européens peut être trouvée dans [Rolling, 1993]. Le type de programme et les partenaires sont énumérés dans [Mariani, 1995]. Des informations administratives sur les projets européens peuvent aussi être retrouvées sur le site de la CE : http://cordis.europa.eu/.

[46] CGLEX : Central European Genelex Model.

[47] Le projet européen Eurolang (décembre 1991 - novembre 1994) regroupait 20 partenaires. Le projet est décrit dans [Seite et al., 1992] et visait le développement d'un système de traduction automatique en 5 langues avec 50.000 entrées dans chaque dictionnaire.

[48] Le Backus Naur Form (BNF) est un metalangage permettant de décrire les règles syntaxiques d'un langage de programmation.

du lexique comprennent : un indexeur, un concordancier, un segmenteur, un tagger, et des outils statistiques pour l'extraction d'informations lexicales de corpus.

Le projet ACQUILEX I (juillet 1989 – juin 1992) avait comme but de construire une base de connaissances multilingue à partir des dictionnaires électroniques monolingues (EN, IT, NL) et multilingues (EN-IT et EN-NL). ACQUILEX II (août 1992 – juillet 1995) s'est fait dans la continuité du premier et a ajouté l'exploitation de corpus pour l'acquisition semi-automatique de connaissances. L'un des buts accessoires du projet était aussi de renforcer les liens entre la communauté lexicographique et celle du TAL.

Les expériences de Genelex et de Multilex trouvent leur suite dans le projet européen EAGLES (janvier 1993 – juillet 1995*), Expert Advisory Group on Language Engineering Standards*. Un groupe d'experts européens ayant fait leurs preuves dans des projets impliquant des travaux lexicaux ou faisant partie du réseau de la communauté du TAL s'est constitué. De nombreux groupes de travail ont édité des recommandations qui sont consultables sur le web[49] et qui couvrent le domaine entier du TAL. Pour l'encodage de corpus, on y trouve notamment le CES, *Corpus Encoding Standard*. Les directives établies par EAGLES seront implémentées par des projets plus appliqués avant d'être repris par ISLE qui en fera une contribution directe à la norme ISO et dont les travaux commenceront dès 2001.

Les buts affichés du projet européen Multext[50] (janvier 1994 – mars 1996) tels que décrits dans ([Véronis et Khouri, 1995]) étaient de standardiser des ressources, textes et données linguistiques, et de créer des ressources linguistiques et des outils génériques. Il s'agit de corpus, de lexiques et d'outils pour un très grand nombre de langues couvrant la plupart des familles des langues européennes. L'un des buts affichés est de tester à grande échelle les recommandations de la TEI et de nouer une collaboration proche avec EAGLES. Concernant les corpus, le format CES (*Corpus Encoding Standard*), a été conçu et mis en œuvre dans les corpus produits. En même temps CES est entré dans les recommandations de EAGLES pour l'encodage de corpus. Le format des lexiques produits est le suivant : forme, lemme, description au format Multext. Ce format du lexique est intéressant car il ne correspond nulle part aux recommandations existantes de Genelex ou de Multilex, mais ressemble plutôt à un format très pratique pour l'exploitation de ces données. Les travaux faits sur la façon de décrire des catégories morphosyntaxiques est très remarquable et capitalise ceux de EAGLES et des grands projets précédents. D'après [Véronis et Khouri, 1995] il est illusoire d'avoir un jeu d'étiquettes commun pour les différentes langues car les jeux d'étiquettes sont incomparables. Les travaux de Multext East[51] ont en plus fait apparaître que les jeux d'étiquettes de Multext sont incomplets.

Le projet européen PAROLE[52] (avril 1996 – mars 1997), *Preparatory Action for linguistic Resources Organisation for Language Engineering*, a produit des corpus (au format CES) et des lexiques correspondants pour toutes les langues européennes. Le format des lexiques et des corpus annotés suit les recommandations EAGLES. Les lexiques comprennent 20 000 entrées par langue et sont codés en SGML au format PAROLE. Ils couvrent des informations morphosyntaxiques et syntaxiques. Dans la suite de PAROLE s'exécute le projet complémentaire SIMPLE (avril 1998 – mai 2000), *Semantic Information for Multifunctional*

[49] A l'adresse suivante : http://www.ilc.cnr.it/EAGLES/browse.html.

[50] Multext : Multilingual Text Tools and Corpora. ([Ide et Véronis, 1994]) Voir http://aune.lpl.univ-aix.fr/projects/MULTEXT/ pour d'amples informations.

[51] Multext East (mars 1995 – février 1997) : Multilingual Text Tools and Corpora for Eastern European Languages. Voir http://nl.ijs.si/ME/.

[52] PAROLE/SIMPLE : http://www.ub.es/gilcub/SIMPLE/simple.html

Plurilingual LExica, qui ajoute 10 000 unités sémantiques à toutes les langues utilisant une structure de qualia[53] étendue. [Lenci et al., 2000], l'article qui présente les aboutissants du projet, le présente comme un cadre général pour le développement de lexiques multilingues. Les spécifications linguistiques sont détaillées dans [SIMPLE Consortium, 2000]) : elles combinent les principes définis dans Genelex, Acquilex et appliquent ceux du lexique génératif tel que définis par [Pustejovsky, 1991] et [Pustejovsky, 1995] dans un cadre mis à jour et publié plus tard dans [Pustejovsky, 2001]. Un outil pour la saisie contrôlée a été développé, mais malheureusement, les archives sur le site le décrivant semblent corrompues.

Le développement de WordNet a débuté à Princeton en 1978 sous l'impulsion de George Miller comme une modélisation du lexique mental. Il a rencontré un franc succès dans le monde du TAL, à tel point qu'il existe aujourd'hui des WordNets dans un très grand nombre de langues, recensés sur le site de *The Global WordNet Association*[54]. En Europe, deux projets européens (mars 1996 – juin 1999) ont développé des WordNets dans 7 langues de l'Europe occidentale et centrale ([Vossen, 1998] et [Vossen, 2002]), suivant le modèle de Wordnet ([Fellbaum, 1998]).

ISLE (janvier 2000 – décembre 2002), *International Standards for Language Engineering*, fort d'une implémentation pratique des directives du projet européen EAGLES, a pris le relais de ce dernier projet et établi des liens avec le comité ISO TC37/SC4, nouant ainsi des liens transatlantiques. L'accent est mis sur le lexique multilingue avec des spécifications de MILE, *Multilingual ISLE Lexical Entry*. ISLE a entre autres donné lieu au développement de IMDI, *ISLE Metadata Initiative* ([IMDI, 2001a]). Il s'agit d'un ensemble de métadonnées qui fait concurrence aux métadonnées OLAC (*Open Data Archive Community*, [Simons et Bird, 2003]) initié par le LDC et SIL International[55] et compatible avec le DC (*Dublin Core*). Etabli à peu près en même temps qu'IMDI au début des années 2000, le but d'OLAC était de définir un ensemble normalisé de métadonnées pour des ressources langagières, alors que IMDI vise des ressources linguistiques et tout d'abord des données multimédia. Dans [IMDI, 2001b] les auteurs pointent sur le fait que les initiatives ne s'adressent pas exactement aux mêmes communautés, et mettent en correspondance les deux propositions. IMDI s'adresse à un sous-ensemble de la communauté OLAC, qui lui-même s'adresse à un sous-ensemble de DC. Avec IMDI, le MPI de Nimègue a développé aussi l'infrastructure et les outils pour éditer les métadonnées et rendre accessibles les ressources décentralisées ([Broeder et al., 2001].

4.1.3 Des projets à plus petite échelle

En dehors du contexte de projets européens quelques projets de création lexicale vont fortement influencer la modélisation des connaissances. S'il s'agit de projets à plus petite échelle, leur succès est dû à des résultats tangibles et à un suivi de qualité.

Le EDR, *Electronic dictionary Research* ([Takebayashi, 1993]), est le résultat d'un projet japonais de neuf ans (1986 – 1994, avec une mise à jour des dictionnaires en 2007). Dictionnaire électronique pour un usage computationnel revendiqué, il contient des lexiques et des corpus dans le but de fournir tout le nécessaire pour un traitement automatique du japonais, de l'anglais et de leur traduction. Dans les lexiques, construits à partir des corpus, on

[53] Que l'on pourrait traduire par « propriété distinctive ». Il s'agit des quatre rôles suivantes : *formal, constitutive, telic, agentive*. Pour *pudding*, dans l'ordre : substance, ingrédients, manger, faire. Voir les références de Pustejovski dans le même paragraphe.
[54] http://www.globalwordnet.org/
[55] Anciennement connu comme le Summer Institute of Linguistics

trouve donc des lexiques de formes, de traduction, de concepts, de cooccurrences et de termes techniques, qui sont liés entre eux pour passer par exemple d'une forme à un concept.

Celex est le centre d'information lexicale néerlandais créé en 1985 comme une collaboration entre les centres lexicologiques importants des Pays-Bas. L'une des principales actions de ce centre était la création du *CELEX Lexical Database* [Baayen et al., 1995], qui contient des lexiques dans trois langues : anglais, allemand et néerlandais. Ils décrivent les formes et les lemmes au niveau orthographique, phonétique, morphologique, syntaxique et fréquentiel. Celex fait aujourd'hui partie du MPI de Nimègue qui joue un rôle central en termes d'infrastructure de ressources linguistiques aux Pays-Bas avec notamment le « Browsable Corpus » qui donne un accès unique à des corpus décentralisés à travers les méta-données IMDI.

PolyLex ([Cahill et Gazdar, 1999]) est un lexique multilingue de 3000 mots en trois langues « liées » (anglais, néerlandais, allemand), implémenté en DATR ([Evans et Gazdar, 1989] et [Evans et Gazdar, 1996]), un langage pour le codage de lexique basé sur l'héritage. Le lexique prend en entrée les lexiques CELEX et construit le lexique multilingue par des techniques d'importation semi-automatiques. L'intérêt direct pour le TAL semble réduit d'après l'auteur même dans [Cahill, 2001]: « *the resulting lexicons themselves are probably not suitable for use in any NLP applications in their present form, due to their rather abstract nature* ». L'idée de mutualiser des informations communes est cependant excellente car l'absence de répétition limite le risque de l'erreur.

Le but du programme DoBeS[56], *Dokumentation Bedrohter Sprachen*, est de documenter un maximum de langues en voie d'extinction dans une démarche de linguistique de terrain. Dans le cadre de ce programme, [Wittenburg, 2001] a passé à la loupe les modèles lexicaux de six langues du projet, cinq autres modèles existants et quatre propositions de modèle. Il adopte une approche pragmatique en laissant de côté tous les modèles qui ne sont pas ceux de DoBeS. Il les représente graphiquement et en déduit un schéma générique à des fins de documentation. Ce schéma sera implémenté dans la base de données qui reçoit les données des langues du monde entier. Il s'approche ainsi un peu de l'ALM, *Abstract Lexicon Model*, qu'il présente au début de l'article comme la définition générique des catégories d'objets lexicaux, leurs caractéristiques et leurs relations. Ces catégories sont les briques (complexes) d'un lexique computationnel et représentent des concepts linguistiques pertinents, leurs attributs et les méthodes qui permettent d'y accéder. Il n'y a qu'un pas de la reprise et d'une généralisation de ces travaux au modèle qui sera proposé pour les travaux à l'ISO. [Wittenburg et al., 2002] prépare le terrain et mentionne déjà UML (et RDF) comme le langage qui pourrait représenter cet ALM, qui sera effectivement le choix du comité ISO bien plus tard.

Bien avant ces derniers travaux, [Sérasset, 1993] avait dressé l'état de l'art des structures lexicales. Dans sa thèse [Sérasset, 1994], Gilles Sérasset propose un système universel de gestion de bases lexicales multilingues. Il montre comment définir une base multilingue en définissant l'ensemble des dictionnaires du système et leurs architectures linguistiques. Pour l'architecture logicielle, il fait la distinction entre les niveaux présentation (visualisation), interne (manipulation de l'information linguistique) et base de données (stockage). Cette architecture a été retenue dans Multilex, qui a aussi défini un ensemble d'outils standards pour un système de gestion de bases lexicales multilingues : éditeur, navigateur, vérificateur de cohérence, défauteur, import/export. Les travaux de cette thèse trouveront une suite dans le projet Papillon [Mangeot et al., 2003].

---

[56] Voir http://www.mpi.nl/DOBES pour une description complète.

4.1.4 Une norme ISO qui s'occupe de tout… ou presque

Début des années 2000, les travaux sur l'abstraction des structures lexicales et la recherche d'une norme internationale sont omniprésents dans la communauté qui s'est construite durant l'histoire de généralisation décrite dans 4.1. [Ide et al., 2000] introduit un modèle formel pour décrire la structure d'un dictionnaire électronique. Les auteurs estiment que pour développer un modèle concret et général des dictionnaires, il est essentiel de distinguer le modèle formel du format de l'implémentation. Cette idée va être essentielle dans les travaux de l'ISO par la suite. [Wittenburg et al., 2002] cherche à construire un *Abstract Lexicon Model* à partir des structures lexicales des grands projets précédents et [Monachini et al., 2003] va jusqu'à proposer une norme avec la synthèse des spécifications des projets EAGLES, SIMPLE et ISLE (avec MILE). Les auteurs y font plus particulièrement appel à deux notions de MILE qui seront essentielles pour l'ISO : MLC, *Mile Lexical Classes*, et leurs instanciations, les MDS, *Mile Data Categories*.

En France, la longue histoire de généralisation de structures lexicales continue fin 2001 à travers l'action nationale INRIA Syntax qui se poursuivra jusque fin 2005. C'est dans le cadre de cette action que l'INRIA va capitaliser ses expériences en édition de ressources lexicales.

L'ISO a validé en août 2001 la création du sous-comité TC37/SC4, appelé « Gestion des ressources linguistiques » au sein du TC37 « Terminologie et autres ressources langagières et ressources de contenu ». Ce sous-comité, présidé par Laurent Romary, est donc dédié à la normalisation dans le domaine des ressources linguistiques. Pendant l'été de 2003, une proposition américaine ouvre le nouveau projet *Lexical Markup Framework* auprès du TC37/SC4 avec une proposition annexe pour une représentation des catégories de données (ISO 12620).

La contribution française a d'abord été financée au niveau national par le projet Normalangue-RNIL qui a commencé début 2003 pour une durée de 36 mois. Ce projet a réuni un grand nombre d'acteurs français, académiques comme industriels (dont Sinequa), pour mettre en place un groupe miroir français au TC37/SC4. Les travaux de ce comité d'experts a abouti à la rédaction d'un document de travail ([Francopoulo, 2004]), résultat de longs échanges sur la liste de discussion « Lexiques pour le TAL » qui faisait intervenir environ 70 personnes. Fort de ces travaux, l'apport français à LMF a été rapide et conséquent.

Ce document de travail explicitait que les dictionnaires visés par la norme alors en préparation sont ceux qui sont exclusivement destinés aux applications du traitement automatique de la langue, et excluent expressément les dictionnaires éditoriaux destinés à la lecture humaine.

En tant que participant au projet français Normalangue-RNIL que nous avons rejoint en cours de route, et ensuite comme observateur dans le projet européen Lirics[57] qui finançait l'élaboration de la norme ISO au niveau international, nous avons pu suivre de près la préparation de cette norme. La communauté de chercheurs impliquée dans son édition est assez impressionnante et couvre le monde entier : la dernière liste d'experts[58] sur le site web du TC37/SC4 mentionne 89 personnes différentes sur l'ensemble des projets de la norme.

Voici un petit schéma qui montre la structure des travaux réalisés. Onze projets sont regroupés dans quatre groupes de travail. Le nombre d'experts par projet est indiqué à la fin de chaque ligne.

---

[57] Sinequa faisant partie du groupe de consultation industriel (*Expert Advisory Group*), nous avons pu participer à deux réunions pour exprimer les besoins industriels.

[58] Consultable publiquement sur le site web du TC37/SC4 (http://www.tc37sc4.org), document N052 rev04 du 10 janvier 2008.

WG 1 (Basic descriptors and mechanisms for language resources)

24610-1	FSR (Feature Structure Representation)	1
24610-2	FSD (Feature Structure Declarations)	35
24612	LAF (Linguistic Annotation Framework)	2
	CitER (Citation of Electronic Resources)	1

WG 2 (Representation schemes)

24611	MAF (Morphosyntactic Annotation Framework)	3
24614-1	WordSeg 1 (Word Segmentation)	5
24614-2	WordSeg 2 (Word Segmentation)	5
24615	SynAF (Syntactic Annotation Framework)	1
24617-1	SemAF/Time (Semantic Annotation Framework)	34

WG 3 (Multilingual information representation)

24616	MLIF (MultiLingual Information Framework)	24

WG 4 (Lexical resources)

24613	LMF (Lexical Markup Framework)	7

Comme l'indique le site de l'ISO, LMF, cadre de balisage lexical, a atteint le stade de IS (International Standard) et est donc publié comme norme internationale (le 17/11/08). Désormais on peut y faire référence par ISO 24613:2008. L'autre norme qui a été publiée dans la même série est l'ISO 24610-1:2006 (*feature structure representation*) mais est marquée « à réviser ». N'oublions pas de mentionner aussi la norme ISO 12620:1999 qui est également marquée « à réviser ». Elle règle notamment le registre des catégories de données (DCR, *Data Categories Registry*). Originalement publiée pour le registre des catégories de données terminologiques, elle inclura aussi celles des descriptions linguistiques. Le LORIA étant impliqué dans le projet, le lexique Morphalou est utilisé pour tester le formatage de données au niveau morpho-syntaxique [Romary et al., 2004].

Grâce aux grand nombre d'intervenants, l'adhérence à la norme LMF par la communauté scientifique semblait acquise même avant sa publication. Ainsi, nous avons compté plus de 20 articles qui mentionnent LMF, ISOCAT ou MAF à LREC 2008. Le format n'étant pas normatif, les ressources produites ne sont néanmoins pas aussi facilement réutilisables qu'on ne le souhaiterait.

4.2 Quelques modèles de structures lexicales sous la loupe

L'adoption d'une norme de codage demande un grand investissement en recodage des données existantes. En contrepartie, il n'existe pas de gain immédiat. Au contraire, l'investissement peut aller très loin si on veut être nativement compatible avec la norme pour supprimer tout coût d'intégration. Il faut alors adapter toutes les procédures d'entrée/sortie des outils usuels et changer les habitudes de codage des linguistes manipulant les lexiques. On ne réalisera des économies en coûts d'intégration que si toutes les ressources produites sont au même format et que ce format est stable sur une longue durée.

Les modèles que nous décrivons ci-dessous peuvent tous être critiqués sur l'un des points suivants : le nombre de langues étudiées et couvertes, la complexité du modèle, la complexité ou la verbosité de l'encodage proposé nécessitant des outils d'édition avancées, la correspondance aux besoins du TAL, l'adoption par la communauté. A titre d'exemple, [Baroni et al., 2004] indique que le jeu d'étiquettes italien proposé par EAGLES ne convenait

pas pour annoter le corpus *la Repubblica*. La distinction entre certaines étiquettes n'étant pas fondée sur des propriétés distributionnelles de la langue, ce qui dégradait les résultats du tagger.

Le chemin parcouru n'a pourtant pas été inutile : la norme ISO est simple et modulaire, et normalise le vocabulaire employé. Elle ne propose néanmoins pas de format ni de syntaxe d'encodage, ce qui risque d'entraver le rêve de la réutilisation facile.

Dans ce qui suit, nous examinons les structures lexicales telles que proposées dans quatre projets qui ont marqué l'histoire lexicale du TAL avant de finir sur la norme ISO.

4.2.1 Genelex

Les principes de Genelex sont expliqués dans l'introduction de [Genelex, 1994]. Genelex est un modèle sémantique de données par contraste avec un modèle physique. Il est donc indépendant de l'implémentation. La structure est plate et indépendante d'une quelconque théorie, pour que toute application puisse trouver les données nécessaires dans le lexique par des règles formelles de traduction ou de déduction. Le modèle conceptuel s'exprime ensuite par un modèle relationnel (schémas entité-attribut-relation) avec des commentaires en langage naturel, et le modèle physique par une DTD SGML[59].

Genelex présente trois couches : morphologique, syntaxique et sémantique. Les informations de chaque entrée sont comprises dans des unités, qui s'articulent de la façon décrite dans la figure 9, provenant de [Genelex, 1995].

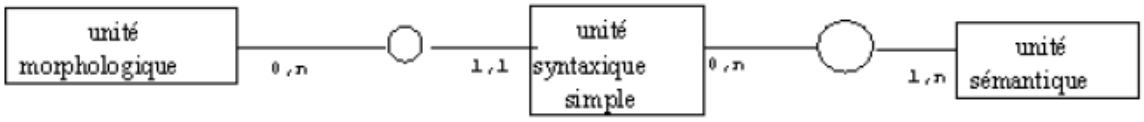


Figure 9 : Articulation entre les différentes couches dans Genelex

Le modèle pour les unités syntaxiques complexes est légèrement plus complexe, mais reste conceptuellement le même : les couches ne sont pas toutes liées entre elles. L'unité morphologique n'a pas de lien direct avec l'unité sémantique. D'après le schéma de figure 9, une unité lexicale est donc soit (1) la combinaison d'une unité morphologique, d'une unité syntaxique et d'une unité sémantique, soit (2) la combinaison d'une unité morphologique et d'une unité syntaxique, soit (3) une unité morphologique. Une exception est faite pour la sémantique des affixes comme on voit dans la figure 10 provenant de [Genelex, 1995].

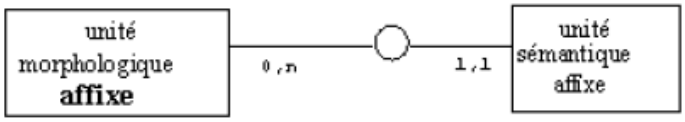


Figure 10 : Articulation entre les couches morphologique et sémantique dans Genelex

[59] Les règles de traduction du modèle conceptuel vers la DTD SGML sont exprimées dans [Genelex, 1993].

La description Genelex est la conséquence du *principe de réalisme linguistique* : « *la stratégie de rechercher la plus grande isomorphie – ou la moins grande distance – entre les objets linguistiques et les objets (entités) du modèle conceptuel de données, sans se préoccuper du modèle physique de données*. » Le modèle Genelex veut donc rester au plus proche du fonctionnement de la langue en ce qui concerne la structure conceptuelle des données. Il respecte également le principe de non redondance des informations. Selon ses spécifications, les informations syntaxiques sont liées aux informations morphologiques, et la sémantique passe par une interprétation syntaxique : certains mots n'ont pas la même signification selon le cadre syntaxique : le verbe *arriver* n'a pas la même signification selon qu'il est transitif direct (*Il arrive à Paris*) ou indirect (*Il arrive à comprendre*).

En suivant le principe de réalisme linguistique tel que modélisé par Genelex, toute analyse sémantique devrait être précédée d'une analyse syntaxique. Or il existe des traitements qui sautent l'étape d'analyse syntaxique et mettent directement en relation les descriptions morphologique et sémantique. Ce fonctionnement n'entre pas dans le modèle Genelex.

Bien que Genelex soit un projet européen, tous les exemples donnés sont en français et aucune référence n'est faite concernant l'applicabilité à d'autres langues.

4.2.2 Multilex

MLEXd ([Paprotté et Schumacher, 1993] propose une norme, écrite en BNF, pour l'encodage de lexiques bilingues. Les langues prises en compte sont : anglais, allemand, français, espagnol, italien, néerlandais et grec. L'inclusion de la problématique de traduction automatique mène à une réflexion plus riche au niveau de l'encodage lexical multilingue. En effet, des mots apparemment simples dans une langue se traduisent par des collocations comme le montrent les deux exemples suivants.

enseignement professionnel
- DE berufsbildender Unterricht
- NL beroepsonderwijs

enregistrement de données
- DE Datenerfassung
- NL vastleggen van informatie

Orienté vers une famille de grammaires (GB, GPSG, HPSG, etc.) sans s'inscrire dans une grammaire spécifique, le formalisme utilisé pour l'encodage de lexique est la structure de traits typés, qui est un formalisme neutre par rapport aux grammaires visées. La plupart des contraintes sont implicites dans la hiérarchisation du lexique, mais certaines contraintes doivent en plus être spécifiées dans une section spéciale. Ces contraintes sont de type : *aucun nom ne peut comporter…* ou *seulement les noms féminins peuvent comporter…*

MLEXd renonce à l'encodage par extension des mots-formes et motive cette décision par le gain en généralisation, la non redondance des informations et la diminution considérable de la taille du lexique.

Il s'agit d'une architecture modulaire qui contient les niveaux suivants : orthographique, phonétique, morphologique, syntactique, sémantique et administratif (informations sur la date de création, l'origine, type et date de modification, statut, etc.). Les deux premiers niveaux traitent les renvois vers les variantes. La structure d'un lexique MLEXd est illustrée dans la figure 11. Les parenthèses indiquent que l'élément peut apparaître une ou plusieurs fois, alors que les accolades indiquent qu'il peut apparaître zéro ou plusieurs fois.

```
body = (gpmu) (word-sense) {phonology-item} {orthography-item}
    {morphology-item} {gpmu-cross-ref-item} (syntactic-item)
    (semantic-item) {lu-cross-ref-item} {bilingual-transfer-item}
    (maintenance-item)
```

Figure 11 : Détail de la structure d'un lexique Multilex

Chaque niveau a son ensemble d'entrées qui communiquent par un système de références. Le gpmu (g*raphic-phonologic-morphological unit*) contient des références vers les informations graphiques, phonologiques et morphologiques et le *word-sense* vers les informations sémantiques en relation avec un ou plusieurs gpmu.

4.2.3 Celex

Comme nous pouvons lire dans [Wittenburg et al., 2004], il était décidé de construire les lexiques Celex en utilisant un système de gestion de base de données (SGBD) dès la soumission du projet. Au contraire des idées reçues des linguistes, le modèle relationnel a pu être défini sans trop de problèmes en coopération avec des experts de base de données de la TU Eindhoven. Cette phase initiale a spécifié le modèle relationnel pour les trois langues : anglais, néerlandais et allemand. Le modèle du néerlandais est composé d'environ 20 tables. La base contient des informations morphosyntaxiques et fréquentielles. Une fois le projet terminé, une sortie en fichier texte tabulé a été réalisée pour que le lexique soit facilement accessible sur un support CD-ROM. L'entrée lexicale est le mot-forme.

L'une des motivations pour utiliser une base de données était la facilité de collaboration. Il s'agissait d'un projet qui a mobilisé huit personnes sur cinq ans qui ont toutes travaillé sur les mêmes données. Sans l'aide d'un SGDB, il aurait été quasiment impossible de garantir l'intégrité du lexique.

A la fin de l'article, les auteurs se prononcent en faveur de XML comme encodage pour les lexiques. De nombreux lexiques sont actuellement produits dans des bases de données aux formats propriétaires (par exemple Microsoft Access) parce que les linguistes sont désormais familiers avec ce genre d'outils, mais la conservation de ces données pourra s'avérer difficile sur la longue durée.

4.2.4 WordNet et EuroWordNet

WordNet ([Fellbaum, 1998]) a originalement été créé à Princeton pour des recherches en psycholinguistique. Il s'agit d'un réseau où les nœuds sont des *synsets*, des ensembles de mots interchangeables dans certains contextes, et les liens sont des relations sémantiques (hyponymie, méronymie, etc.). EuroWordNet ([Vossen, 2002]) est le projet européen qui a créé un WordNet multilingue dans les langues suivantes : néerlandais, espagnol, italien, français, allemand, tchèque et estonien. Chaque WordNet a été créé à l'image du WordNet original, suivant la même structure interne de synsets liés par des liens sémantiques.

L'interlinguisme est réalisé en liant chaque synset de chaque WordNet au synset le plus proche dans le WordNet de Princeton. Les synsets de ce dernier fonctionnent ainsi comme un index multilingue, nommé ILI, *Interlingual Index*. Cet index est structuré par une ontologie.

Les BC, *Base Concepts* sont également des éléments structurants. Ce sont les synsets les plus importants de chaque WordNet. Elles sont sélectionnés par rapport au nombre de relations

avec les autres synsets et leur position plus ou moins élevée dans la hiérarchie ontologique. Ils sont mis en relation entre les langues, et forment une grille multilingue.

4.2.5 La norme ISO 24613:2008

La norme ISO[60] décrit un métamodèle pour l'encodage lexical. Le modèle est représenté par un diagramme de classes en UML, où les classes représentent des objets lexicaux et les associations lient ces mêmes objets. Le cœur du modèle est obligatoire et les extensions sont optionnelles. Comme on voit dans la figure 12, le cœur du modèle représente une ressource lexicale qui peut être faite d'un ou de plusieurs lexiques, qui contiennent une ou plusieurs entrées lexicales. Une entrée lexicale contient au moins un objet de type *Form* (= un lexème), et possiblement un ou plusieurs *Sense*.

Les attributs des objets ne sont pas indiqués sur les schémas. Les attributs et leurs valeurs sont appelés catégories de données et enregistrés dans le DCR, le *Data Category Registry*, consultable sur le web[61]. L'autorité de registration est le MPI de Nimègue qui développe et héberge l'infrastructure. La normalisation de la description linguistique passe donc par une centralisation et uniformisation du vocabulaire employé. L'édition du DCR est affaire de comités d'experts qui gèrent les entrées (ajout, validation, etc.). Il est possible de l'enrichir avec des catégories de données utilisateur. Un DCS, *Data Category Selection*, est l'ensemble des catégories de données utilisé dans un lexique LMF.

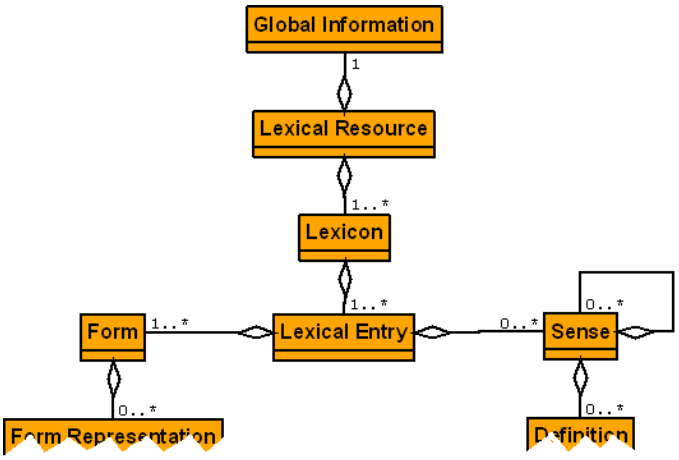


Figure 12 : Représentation partielle du cœur du modèle LMF (*Core Package*)

Un lexique correspondant à la norme LMF est un lexique qui respecte le cœur de LMF, emploie un DCS dont les catégories de données sont enregistrées dans le DCR, et peut contenir des extensions qui répondent aux mêmes exigences que le cœur. Les extensions sont : morphologie, dictionnaire électronique, syntaxe, sémantique, notations multilingues, motifs morphologiques, motifs d'expressions multi-mot et expression de contraintes. La différence entre les extensions morphologie et motifs morphologiques est que la première prévoit une description en extension et le second en intension. Le modèle de chaque extension

[60] Nos informations proviennent du document [ISO-TC37/SC4, 2008] qui était la dernière soumission avant approbation finale. Pour des raisons de copyright, nous avons refait le schéma et n'en présentons qu'une partie.
[61] http://www.isocat.org/

est normatif, et est accompagné d'une instanciation informative qui comporte parfois des exemples en XML. Aucun format n'est donc imposé et la norme reste purement conceptuelle.

A titre d'exemple, l'extension *morphology* est (partiellement) donnée en figure 13. Par convention, les nouvelles classes sont colorées alors que celles qui font partie du cœur du modèle sont blanches.

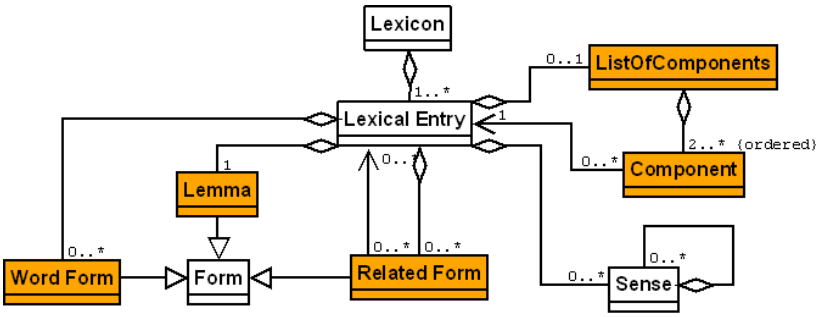


Figure 13 : Représentation partielle de l'extension *morphology* de la norme LMF

On voit notamment que si on adopte cette extension le lemme définit l'entrée du lexique. Sa relation unique avec l'entrée lexicale l'emporte sur la multiplicité de la forme qui était définie dans le cœur du modèle. Il est indiqué que le lemme est choisi par convention, mais il n'est pas prévu de pouvoir expliciter la règle. Dans cette extension on prévoit également l'énumération ordonnée des composants d'expressions multi-mots (dont les objets sont aussi utilisés dans d'autres extensions).

Tout comme l'extension *morphology* se construit autour de l'objet *Form*, l'extension *semantics* se centre autour de *Sense*. Elle intègre la sémantique de type WordNet par un objet *Synset* en rassemblant une ou plusieurs *Sense*, mais également une sémantique basée sur les rôles syntaxiques qui s'articule avec l'extension *NLP syntax*.

4.3 Des structures lexicales à une architecture linguistique

La vue adoptée jusqu'ici est purement lexicale et elle s'arrête aux frontières du lexique. Le terme d'*architecture linguistique* pour désigner un modèle lexical fait son apparition au début des années 90 dans les travaux de EAGLES et dans la thèse de Gilles Sérasset. Ce dernier s'inspire des modèles de Genelex et de Multilex pour mettre au point un système d'édition lexicale d'un dictionnaire multilingue multi-applications. Eagles a écrit ses directives à partir des travaux faits dans les projets Genelex et Multilex.

Ils proposent les définitions suivantes :

> [EAGLES, 1993][62] : « L'architecture linguistique définit les objets élémentaires du modèle et leurs relations. Elle spécifie aussi la terminologie générale commune au standard complet et utilisée pour discuter des dictionnaires, de leurs composants ou de l'interaction entre eux »

[62] "The linguistic architecture defines the basic objects of the model and their relations. It also specifies the general terminology which is common to the whole standard and used to talk about the dictionaries, their components and the interaction of these." (http://www.ilc.cnr.it/EAGLES/lexarch/node5.html)

> [Sérasset, 1994] : « Organisation des informations linguistiques des unités des dictionnaires » (résumé de la thèse) « comment les entrées sont-elles organisées » (p. 51) « L'architecture linguistique définit les objets de base d'un dictionnaire et leurs relations. » (p. 52). « L'architecture linguistique d'une base lexicale définit la manière dont sont codées les entrées des dictionnaires qu'elle contient. Elle régit donc, notamment, les structures logiques qui seront utilisées dans le codage des informations linguistiques. » (p. 60)

Ces définitions sont issues de travaux sur la généralisation de structures lexicales et se limitent au lexique. Axées sur les travaux lexicaux, elles laissent totalement de côté l'existence d'informations dans les autres ressources. Or une partie des connaissances lexicales peuvent être codées dans des règles pour traiter les mots hors vocabulaire. D'une part un lexique n'est jamais exhaustif et d'autre part, une plus petite taille du lexique facilite la gestion. Au traitement très bas niveau de projection du lexique sur le flux textuel, le lexique s'articule donc avec des règles lexicales qui font entièrement partie du modèle. Ce modèle n'est alors plus totalement *lexical*, mais devient un modèle des informations véhiculées dans un traitement. Ainsi nous replaçons le lexique dans son contexte original de ressource linguistique servant à un traitement. Il est une ressource comme les autres. L'architecture linguistique doit donc être établie sur l'ensemble des ressources prises en compte par le traitement, que ce soit des lexiques, des grammaires ou des corpus.

Comme les ressources, le traitement dispose également d'une architecture linguistique. A ce niveau de description conceptuelle, ressources et traitement doivent partager une partie du modèle de connaissances : les informations dont a besoin le traitement sont un sous-ensemble des informations qui existent dans les ressources. Si le traitement exploite toutes les informations, ils partagent le même modèle avec les ressources.

Nous reprenons donc les définitions mentionnées ci-dessus pour les étendre à l'ensemble des informations (nommés objets ci-dessus) linguistiques utilisées dans un système, tout en excluant les références à l'organisation physique des ressources.

Ces informations étant considérées dans le cadre d'un traitement influant sur le processus en cours, nous pouvons les considérer comme des connaissances.

> Une *architecture linguistique* est un modèle qui représente un ensemble de connaissances linguistiques ainsi que les relations entre ces connaissances.

Ce modèle est abstrait et ne tient pas compte de l'organisation physique des ressources, que ce soit le type ou la syntaxe d'encodage.

Une ressource linguistique tout comme un traitement en TAL dispose implicitement d'une architecture linguistique. L'architecture linguistique d'un système est la synthèse des architectures linguistiques de toutes les ressources et de tous les traitements du système. Pour dresser l'architecture linguistique d'un système et avoir une vue globale sur toutes les connaissances véhiculées, il faut donc d'abord définir l'architecture de chaque traitement et de chaque ressource.

Nous nous sommes servis de la modélisation de l'architecture linguistique pour obtenir une vision claire de haut niveau sur le fonctionnement du système qui est étudié. Ce modèle était un préalable à la conception d'un outil de gestion et a servi de modèle de référence pour valider des choix techniques et pour comprendre certains dysfonctionnements. Quand les ressources sont éclatées sur un grand nombre de fichiers, une modélisation au niveau du système est obligatoire pour obtenir la vision nécessaire pour écrire des outils de gestion efficaces.

[Bonhomme et Lopez, 2000] ont recouru au même principe pour rassembler dans une seule ressource en XML trois types de ressources dans le cadre de travaux sur le formalisme de grammaire LTAG : un lexique morphosyntaxique, un lexique syntaxique et des schémas d'arbres. Comme un codage direct en XML n'était pas satisfaisant, ils ont utilisé un modèle formel de représentation abstraite des informations linguistiques contenues dans les ressources : RROM, *Relational Resource Organisation Model*. Un RROM est composé d'un ensemble d'entités de ressources (RE, *Resource Entity*) et d'un ensemble de relations entre ces entités. Une RE est une catégorie de données indépendante et abstraite utilisée en TAL (comme par exemple mot, lemme ou catégorie grammaticale) et est représentée graphiquement par un rectangle. Son instanciation est représentée par un ovale. Les relations (RR, *Resource Relation*) ont un couple d'arités qui indiquent l'arité de la relation en fonction de la direction de cette relation. Après une brève illustration sur le lexique Multext, les auteurs dressent le RROM des trois ressources susmentionnées. Nous reprenons ce modèle dans figure 14 qui, comme les auteurs l'indiquent, ne contient pas les schémas d'arbres ni la représentation des traits pour des raisons de compréhensibilité. La partie supérieure du modèle est la partie morphosyntaxique, au milieu se trouve la partie syntaxique et la partie inférieure est la partie sémantique. Verticalement au centre se trouvent les instanciations, par exemple *lemma representation* pour *lemma* et *inflection* pour *flexed word*. Ces instanciations nous semblent nécessaires pour la conception du format XML qui est déduit directement de cette formalisation. Pour notre modélisation, l'inclusion de ces instanciations est superflue : toute information recensée peut être instanciée.

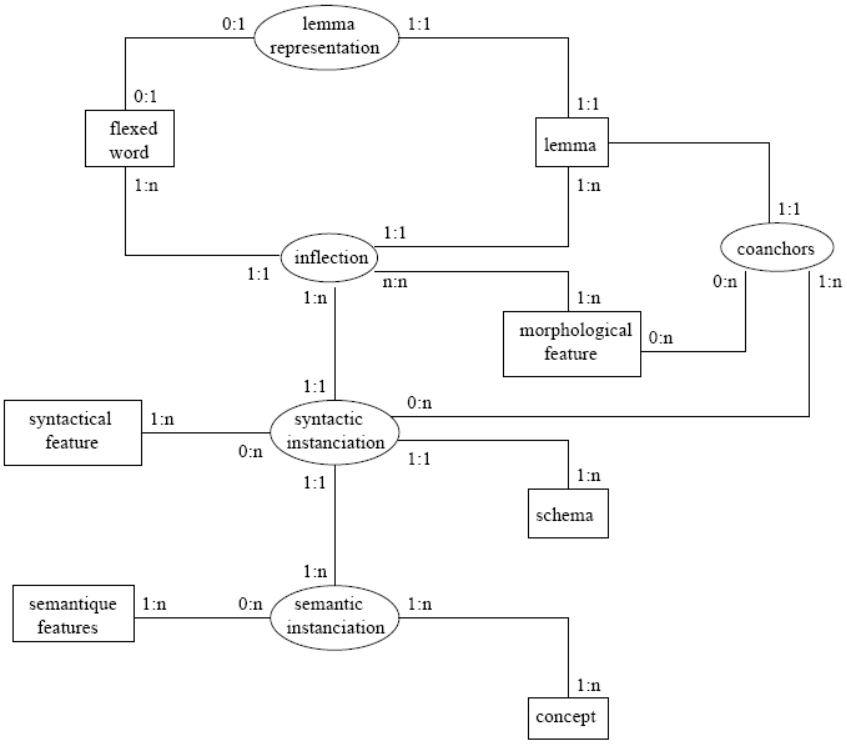


Figure 14 : RROM des ressources LTAG

A part une compréhension totale des informations véhiculées, nous voyons au moins deux autres applications directes possibles de la description de l'architecture linguistique. Elle peut tout d'abord servir pour normaliser le vocabulaire des API des applications de TAL de manière à faciliter les échanges entre des modules logiciels indépendants ou les

enchaînements dans des chaînes de traitement comme par exemple celles respectant le cadre UIMA. Ensuite, dans un monde où l'échange des ressources est chose commune, les ressources et les traitements devraient expliciter formellement leurs architectures linguistiques pour examiner leur compatibilité. Ainsi, producteur de ressources et utilisateur peuvent se retrouver par l'intermédiaire d'une modélisation des connaissances commune.

4.4 Conclusion

Après une longue histoire de travaux de normalisation, une norme ISO a été établie sur l'encodage des informations linguistiques dans les lexiques pour le TAL. A l'origine de la normalisation se trouve la volonté de réutiliser les lexiques et de réduire leur coût d'adaptation. Au cours de ces travaux est apparu le terme d'*architecture linguistique,* pour désigner les objets lexicaux et les relations entre eux. Nous avons étendu cette notion à l'ensemble des connaissances contenues dans les ressources linguistiques : lexiques, corpus et grammaires. L'architecture linguistique joue un rôle central dans nos travaux. Elle modélise la façon dont les connaissances sont exploitées par les traitements et procure une compréhension profonde du fonctionnement du système. L'architecture linguistique est indispensable pour façonner les outils de gestion des ressources.

PARTIE 3

Modélisation d'une architecture linguistique

Chapitre 5

Identifier les traitements et leurs connaissances

Pour pouvoir concevoir les outils de gestion, nous devons décrire l'architecture linguistique du système, c'est-à-dire expliciter les liens entre les connaissances, les ressources et les traitements. La méthode que nous proposons dans ce chapitre est de séparer fonctionnellement les traitements et de décrire les besoins en connaissances de chaque traitement avant de modéliser les informations correspondantes dans les ressources. Nous examinons de façon générale les traitements habituellement utilisés en RI pour répondre aux défis linguistiques que pose le traitement de la langue. Selon la méthode mise en œuvre, ces traitements peuvent reposer plus ou moins sur des connaissances linguistiques. Le découpage en traitements fait partie du modèle et reflète en cela déjà notre vision. Ce chapitre se veut générique et ne décrit pas les ressources linguistiques, ce qui demanderait de choisir une méthode pour chaque traitement. Cela est fait dans le chapitre suivant en prenant l'exemple concret des traitements et ressources de l'analyse linguistique de Sinequa. Ce chapitre fournit donc le cadre général avant l'application concrète.

5.1 Méthode de modélisation

Pour pouvoir concevoir les outils de gestion, nous devons modéliser les connaissances contenues dans les ressources linguistiques. Or, ces ressources peuvent contenir un grand nombre d'informations, dont certaines ne sont pas ou plus exploitées par les traitements. Il est alors nécessaire d'identifier les connaissances requises par les traitements afin de connaître exactement la relation entre la ressource et le traitement qui l'exploite.

Dans une chaîne de traitement, les traitements ne sont pas toujours nettement séparés. En effet, pour des raisons d'optimisation, ils peuvent s'imbriquer au niveau du code. Le découpage que nous proposons est donc fonctionnel. Dans des chaînes moins optimisées que sont souvent les plateformes, notre découpage peut très bien correspondre à des briques logicielles qui se limitent strictement à leur fonction.

Les traitements que nous avons identifiés sont utilisés en RI pour résoudre certains problèmes qui sont d'origine linguistique. Il existe souvent des méthodes concurrentes, et il nous semble important de présenter leurs avantages et/ou inconvénients ainsi que les performances trouvées dans l'état de l'art. En effet, certaines de ces méthodes s'appuient plus sur des ressources linguistiques que d'autres avec des divergences de performance. Cela aidera à situer les méthodes appliquées par Sinequa pour lesquelles nous ne pouvons diffuser de chiffres évaluatifs.

L'ordre dans lequel nous présentons les traitements est l'ordre logique d'exécution des traitements. Certains traitements produisent les informations que d'autres vont prendre en entrée.

Nous décrivons pour chaque traitement son objectif, les méthodes existantes pour accomplir cet objectif et les connaissances nécessaires pour exécuter chaque méthode.

Les informations linguistiques prises en compte sont référencées en faisant abstraction de l'encodage physique des ressources. Peu importe donc si les informations sont éclatées sur un grand nombre de fichiers ou centralisées dans une structure plus ou moins complexe.

Les traitements sont symbolisés par le caractère θ. Pour chaque traitement, nous décrivons l'entrée et la sortie, puis les connaissances utilisées lors du traitement. Pour des raisons de clarté, les symboles représentant les connaissances peuvent avoir plusieurs lettres. Ils commencent par des minuscules alors que les ensembles sont représentés par des majuscules.

La méthode que nous proposons pour construire le modèle se déroule selon les étapes suivantes :

1. séparer fonctionnellement les traitements ;
2. décrire les entrées et sorties des traitements ainsi que leurs besoins en connaissances ;
3. décrire les informations contenues dans les ressources.

Nous parcourons les deux premières étapes dans ce chapitre de façon générale. Nous utiliserons les résultats comme cadre pour modéliser l'analyse linguistique de Sinequa dans le chapitre suivant.

5.2 Identification de la langue

Objectif

Afin de pouvoir appliquer la grammaire de découpage correspondant à la langue du texte et de découper celui-ci en unités textuelles, un système traitant une base documentaire multilingue avec des documents monolingues ou multilingues doit identifier la langue du document ou des parties du document pour les documents multilingues.

Méthodes

Les méthodes d'identification de la langue existantes agissent au niveau des lettres (n-grammes), ou des mots en s'appuyant sur les mots fréquents ou les mots grammaticaux. Avec des résultats en classification de documents allant jusqu'à 99,8 % ([Cavnar et Trenkle, 1994]), on pourrait imaginer qu'il s'agit d'une affaire réglée. [Prager, 1999] montre néanmoins qu'il faut une taille minimale du document pour obtenir un score de 100 % et que les langues similaires comme les paires bokmål/danois et catalan/espagnol sont plus difficiles à distinguer. [Langer, 2002] a dressé la liste des problèmes qui restent à surmonter, dont les documents courts (< 10 mots), la présence d'une grande quantité de noms propres (par exemple une liste de noms de vins rouges dans un document allemand), et la non classification d'un document dont le système ne connaît pas la langue. Il considère les documents multilingues comme un moindre problème, car la langue majoritaire du document est bien reconnue.

Les documents multilingues sont néanmoins présents dans beaucoup d'environnements. Il en existe deux types : des documents avec des citations dans une autre langue que le corps du texte, ou bien des documents avec de grands blocs de textes en langues différentes. Dans ce dernier cas, les parties peuvent être consécutives, par exemple dans un rapport français qui porte sa traduction anglaise dans le même document d'une entreprise internationale, ou bien parallèles, comme dans un appel d'offres de l'état belge. Dans tous les cas, les différentes langues des parties du document doivent être identifiées, et il n'est pas évident de trouver au mot près le passage d'une langue à l'autre, comme on peut lire dans [Mandl et al., 2006] et [Artemenko et al., 2006]. La présentation affichée de ce dernier article fait état d'un taux de réussite pour trouver la frontière du changement de langue à deux mots près qui varie de 81 à 97% selon le type de multilinguisme. La citation semble la plus difficile à repérer correctement. Tout type de multilinguisme confondu, 3 à 6 % des passages sont mal[63] reconnus ou ne le sont pas.

Le taux d'erreur que présente l'identification de la langue ou sa carence en couverture des langues a des incidences sur toute la suite de la chaîne de traitement. Si on applique les traitements, que ce soit la lemmatisation ou la détection des entités avec les ressources d'une autre langue, les résultats sont catastrophiques. Les traitements et leurs ressources doivent être assez robustes pour pallier les erreurs d'une couche plus basse pour empêcher la propagation d'erreurs. Si le système n'est jamais confronté à des documents multilingues, les traitements et ressources n'ont pas à prendre en compte les possibles erreurs et seront donc très différents. Les exigences du système final, en l'occurrence le traitement multilingue de documents possiblement multilingues, façonnent les traitements de tout le système à travers l'adaptation des ressources qu'ils utilisent.

[63] Avec une différence de plus de 4 mots.

Connaissances

La fonction d'identification de la langue θ_{idl} prend en entrée un texte *t* et des connaissances C_{idl}, et donne en sortie une séquence de fragments de texte $frag_i$, chaque fragment étant associé à une langue l_j.

$$\theta_{idl}(t, C_{idl}) = ((frag_1, l_1), (frag_2, l_2), ..., (frag_m, l_n))$$

où $l_i \neq l_j, \forall (i, j) \in [1, n]^2$

et $(frag_1 \bullet frag_2 \bullet ... \bullet frag_m) = t$

où $\bullet$ représente la concaténation des fragments *frag*

Les connaissances C_{idl} utilisées en identification de la langue se limitent aux couples de n-grammes de caractères *ngram* et leurs probabilités *pNgram*, aux ensembles de mots fréquents *mfr* ou de mots grammaticaux *mg* (comme les pronoms, déterminants et auxiliaires) d'une langue *l*.

$$C_{idl} = \{N, F, O\}$$

où $N = \{N_{l1}, N_{l2}, ..., N_{ln}\}$,

$F = \{F_{l1}, F_{l2}, ..., F_{ln'}\}$,

et $O = \{O_{l1}, O_{l2}, ..., O_{ln''}\}$

avec *n, n'* et *n''* étant le nombre de langues,

$N_{li} = \{(ngram_1, pNgram_1), (ngram_2, pNgram_2), ..., (ngram_m, pNgram_m)\}$,

$F_{li} = \{mfr_1, mfr_2, ..., mfr_{m'}\}$

et $O_{li} = \{mg_1, mg_2, ..., mg_{m''}\}$

La langue du fragment étant identifiée, les traitements suivants appliqueront les connaissances de la langue détectée sur des fragments de langue uniforme. Pour ne pas complexifier la notation inutilement, nous ne mentionnerons pas la langue en indice dans la suite de ce chapitre.

5.3 Découpage du flux en unités textuelles

Objectif

Afin de pouvoir analyser un texte, celui-ci doit être découpé en petits morceaux permettant de faire le lien avec la base de connaissances. Un premier découpage basé sur les signes d'espacement et de ponctuation permet d'obtenir des unités simples. Ce découpage peut être entièrement fait par application d'une grammaire. Si les mots-formes de la base de connaissances sont complexes, c'est-à-dire s'ils comportent un tiret ou un espace, la grammaire doit prévoir la recomposition d'unités simples en unités complexes. L'objectif est donc de découper un texte en unités textuelles qui, par appariement, donnent accès aux informations de la base de connaissances.

La présence d'unités complexes dans la base de connaissances complexifie énormément la grammaire de découpage. Ces unités sont nécessaires pour les traitements ultérieurs, et correspondent également à une réalité linguistique. Syntaxiquement et sémantiquement ces

unités complexes disposent d'une autonomie qui est vérifiable par des critères grammaticaux. Il s'agit aussi bien d'unités syntaxiques (comme *à condition que*) que de mots composés. [Gross, 1988] donne un ensemble de critères pour distinguer les *mots composés* des simples suites syntaxiques, dont l'adjonction d'un autre adjectif en coordination pour les suites *nom adjectif*. Ainsi *une étoile minuscule et brillante* est possible, mais pas *une étoile filante et brillante*, ce qui l'amène à conclure que *étoile filante* est un mot composé. Comme l'atteste le même auteur, le figement des mots composés est variable et les uns sont plus figés que les autres. C'est sans doute pourquoi il peut être difficile d'obtenir un consensus sur l'existence de certaines unités complexes.

La sémantique de ces unités complexes n'est pas purement compositionnelle : le sens de l'unité composée n'égale pas la somme des composants. Ainsi une *nature morte* n'est pas une *nature* qui est *morte*. Dans d'autres cas, le sens peut être directement lié, comme dans *plante verte*. Une *plante verte* est bien une *plante* qui est *verte*, mais ce n'est pas la verdure qui lui donne son sens. Sa définition dans le Petit Robert se trouve dans l'entrée *plante* comme suit : *Plantes d'appartement, plantes vertes : plantes décoratives, à feuilles persistantes, qui peuvent croître dans une maison*. La sémantique joue donc un rôle de premier ordre dans le découpage en unités textuelles.

Selon les applications, ces critères peuvent être variables. Ainsi, en terminologie, où les termes sont le plus souvent complexes, la compositionnalité du sens n'est pas un critère. Une *table verte* et une *table carrée* peuvent très bien être des termes d'une terminologie, sans pour autant correspondre à la définition d'un mot composé.

Dans les langues compositionnelles, les grammaires traditionnelles utilisent des critères purement morphologiques pour identifier un mot composé : il suffit que deux unités autrement autonomes se collent, le critère de la compositionnalité sémantique n'est pas pris en compte. Le sens peut être unitaire comme dans le premier exemple suivant, ou être égal à la somme comme dans le second.

DE	*Federbett*	couette	Feder | Bett
	Bildungsgesetz	loi sur la formation	Bildung | Gesetz

En recherche d'information, la distinction entre les deux est très importante, car dans le second cas on veut chercher sur les composants, mais de préférence pas dans le premier cas. La combinaison existe aussi, comme dans l'exemple suivant :

DE	*Berufsbildungsgesetz*	loi sur la formation professionnelle

On ne peut pas raisonnablement traiter la structure interne des mots sans prendre en compte les catégories grammaticales, comme nous verrons dans 5.4.

En recherche d'information comme pour toute application en TAL, le découpage en unités textuelles est extrêmement important. Nous avions par exemple décelé une erreur dans la grammaire du danois, qui, dans certaines circonstances de codage HTML très précises, gardait comme une seule unité le patron « chiffre + espace + token[64] ». En voici quelques exemples :

1 Bygningshistorie
3 Århus
1 Anlægsarbejder
7 Økonomi

[64] Un *token* est une suite de caractères délimitée par une ponctuation, des espaces ou une combinaison des deux.

Dans l'analyse totale du Wikipedia danois[65], ce problème concernait 842 428 tokens (60 587 tokens uniques) qui n'avaient pas été analysés. Dans un moteur de recherche ce genre d'erreur est grave car les unités sont ensuite indexées comme des mots composés, et font chuter la pertinence des documents qui les contiennent selon les lois du TF/IDF. En même temps, dans une utilisation réelle, la chance qu'un utilisateur remarque le silence provoqué est quasi nulle tant qu'il existe d'autres documents qui contiennent les mots en question et qu'il ne cherche pas un document en particulier. Le corpus total contient en effet 130 776 219 tokens.

Méthodes

Le découpage repose tout d'abord sur une grammaire construite autour des signes d'espacement et de ponctuation. Cette grammaire est différente selon la langue. L'un des meilleurs exemples est le point indiquant un ordinal en allemand. Un point suivi d'un mot capitalisé n'est pas forcément un point de fin de phrase mais peut indiquer un ordinal s'il est précédé d'un chiffre, comme l'illustre l'exemple suivant. La règle générale n'est donc pas telle quelle valable en allemand.

DE am 1. Februar = am ersten Februar (le premier février)

Même à l'intérieur d'une langue les signes de ponctuation et d'espacement sont ambigus. En français, un tiret peut par exemple indiquer une incise, un mot composé (par exemple : *tire-bouchon*) ou être purement grammatical (par exemple : *dit-il*). Un point peut indiquer la fin d'une phrase ou faire partie d'une abréviation, voire être les deux en même temps au cas où l'abréviation ou le sigle terminent la phrase, comme pour *etc.* dans l'exemple suivant.

FR […] *dans les grandes institutions publiques, à Tachkent, à Boukhara, à Douchanbé, etc. Les fonds maintenus* […]

Certains signes sont particulièrement ambigus, comme l'apostrophe. Elle est parfois utilisée comme signe fermant une apostrophe culbutée (exemple français), alors que dans certaines langues, elle peut également faire partie d'un mot (exemple anglais), le commencer (exemples néerlandais), ou le terminer (exemple allemand).

FR Ce mémoire est 'excellent'.

EN *St. James's Palace*

NL *Hij drinkt koffie 's morgens.* (Il boit du café le matin.)
's Ochtends drinkt hij koffie. (Le matin, il boit du café.)

DE *Karl Kraus' "Letzten Tagen der Menschheit"*
(Le [livre] "Letzten Tagen der Menschheit" de Karl Kraus)

A cause de l'ambiguïté des signes de ponctuation, les règles de découpage sont rarement absolues et le défi est de trouver la meilleure précision possible. [Grefenstette et Tapanainen, 1994] donnent un bon aperçu des heuristiques utilisées avec des taux d'erreurs calculés sur le Brown Corpus. L'interprétation de tout point comme fin de phrase mène par exemple à la bonne reconnaissance de 93,20% des phrases. L'ajout d'heuristiques plus fines, sans ressources autres que des règles donne 97,66% de bonne reconnaissance. En ajoutant des lexiques en plus, ils arrivent à une bonne reconnaissance de 99,07 % des phrases. Les résultats donnés concernent le seul découpage en phrases, et non pas celui en unités. Comme il peut y

[65] Les dumps statiques en HTML de Wikipedia sont téléchargeables sur http://static.wikipedia.org/. Notre version date du 8 juin 2008.

avoir des erreurs à cette étape, les traitements ultérieurs qui reposent sur le découpage, devront en tenir compte.

Le découpage qui tient compte des mots-formes complexes peut se faire de deux manières, qui ont leurs avantages et inconvénients. Soit on laisse l'ambiguïté de découpage, c'est-à-dire que *pomme de terre* est à la fois considéré comme une et comme trois unités, soit on supprime l'ambiguïté, prenant le mot-forme complexe comme seule unité. L'avantage de la première approche est qu'elle laisse ouverte des interprétations possibles, comme pour *pomme de terre cuite*, qui sera découpé en :

pomme + de + terre + cuite
pomme de terre + cuite
pomme + de + terre cuite

C'est la solution qui a été choisie dans CLAWS ([Leech et al., 1994]) pour l'étiquetage du BNC. L'inconvénient de cette approche générique est que la complexité est exponentielle, ce qui est difficilement conciliable avec les contraintes d'une application industrielle. Dans ce dernier cas, on fait plutôt le choix de l'efficacité en découpant selon les mots-formes du lexique les plus longs. L'ambiguïté entre une suite syntaxique et une unité complexe peut alors empêcher l'analyse comme unité complexe.

Ici encore, les traitements ultérieurs doivent prendre ce choix en compte. Si *pomme de terre* est considéré comme une unité, alors *pomme de terre verte* correspondra au motif *nom adjectif*, et non pas à *nom préposition nom adjectif*, ce qui fait une différence dans le cadre d'une extraction de groupes nominaux par exemple.

Cette façon de traiter les unités complexes a des conséquences directes sur les entrées lexicales. Certaines unités ne peuvent être codées dans le lexique sous peine d'empêcher une analyse correcte. Ainsi la conjonction *bien que* et l'adverbe *d'accord* ne peuvent être codés dans le lexique sous peine d'empêcher toute autre analyse comme illustrée dans les exemples suivants.

FR	bien que	Il voyait bien que le chevalier de Guise était son rival.
	d'accord	protocole d'accord commun

La présence des unités complexes dans les ressources est obligatoire. En effet, un traitement par motif en utilisant le mot-forme des mots, par exemple autour du mot-forme *de* n'identifierait pas que des unités complexes (comme *venu de Paris* dans *Il était venu de Paris*), et ne ramènerait pas ceux composés d'autres motifs (comme *plante verte*). Un traitement par patrons de catégories grammaticales comme *nom 'de' nom* et *adjectif nom* n'est possible qu'en ajoutant les connaissances de mots-formes simples avec leur description grammaticale. L'ajout de la catégorie grammaticale nous amène au problème de l'ambiguïté grammaticale (*plante* peut être verbe ou nom) et le traitement de désambiguïsation, qui est le prochain traitement. A ce niveau, on ne peut qu'ajouter un ensemble de mots-formes complexes dans les connaissances.

Ces unités complexes posent également des problèmes pour l'évaluation de la désambiguïsation morphosyntaxique. Celle-ci passe par la comparaison entre un corpus étiqueté par le tagger et son référentiel, qui est le même corpus étiqueté par une personne, en calculant les mesures de précision et de rappel. Le fait que les unités ne soient pas découpées de la même manière est la principale difficulté pour mesurer la performance d'un tagger : comment prendre en compte que les deux corpus ne contiennent pas les mêmes unités complexes ? L'un des problèmes à résoudre dans l'évaluation de modules de désambiguïsation est le découpage en unités qui peut être différent selon les modules, car

toutes ne recensent pas les mêmes mots composés. Confronté à cette difficulté dans le projet Grace, [Paroubek et Rajman, 2000] proposent un étiquetage minimal de tous les composants, par exemple :

avant[Adv|Sconj/1.2] que[SConj|SConj/2.2].

Ce traitement existe bien dans l'analyseur de Sinequa, mais pour des raisons d'optimisation les connaissances nécessaires ne sont pas codées dans une ressource mais font partie du code source. Pour cette raison, nous mentionnons le traitement dans le chapitre suivant sans qu'il n'y ait une ressource de grammaire ou de lexique correspondant.

Connaissances

Le découpage prend en entrée un fragment de langue unique constitué de caractères *c*, qui sont des lettres et des signes d'espacement et de ponctuation tels que décrits par les normes ISO/CEI 10646[66] et Unicode. La lecture séquentielle des caractères produit un flux textuel qui est découpé en une liste d'unités textuelles *u* en appliquant des connaissances C_{dec}. Le texte et ses unités textuelles correspondent à *A**, *A* étant l'ensemble fini de caractères *c* correspondant au système orthographique de la langue *l*. La langue étant connue, la fonction de découpage θ_{dec} applique à chaque fragment *frag* la grammaire de la langue préalablement identifiée ou définie par défaut.

$$\theta_{dec}(frag, C_{dec}) = (u_1, u_2, ..., u_n)$$

où $frag, u_i \in A*$

avec $A = \{c_1, c_2, ..., c_m\}$

Le découpage se fait selon les règles $rDec_i$ de la grammaire de découpage *Gdec*. La grammaire s'articule autour d'un ensemble *P* de signes d'espacement et de ponctuation p_i, *P* étant un sous-ensemble de *A*. Deux approches existent, l'une dont les règles prennent en compte un ensemble *L* de mots-formes f_i, l'autre qui se contente d'appliquer des règles autour des signes p_i.

$C_{dec} = \{Gdec, P, L\}$ ou $C_{dec} = \{Gdec, P\}$

où $Gdec = \{rDec_1, rDec_2, ..., rDec_n\}$,

$P = \{p_1, p_2, ...p_{n'}\}$, $P_l \subset A$

et $L = \{f_1, f_2, ..., f_{n''}\}$ où $f_i \in A*$

[66] L'ISO/CEI 10646 définit l'Universal Character Set (UCS) qui représente un jeu de caractères où chaque caractères associe une description unique à un entier positif appelé son point de code. La norme, qui interdisait par exemple l'utilisation de caractères de contrôle, a été unifiée avec la norme Unicode au début des années 90, qui l'inclut comme un sous-ensemble de ses spécifications.

5.4 Analyse et étiquetage morphosyntaxique des unités textuelles

Objectifs

Après l'identification de la langue et le découpage, le fragment est représenté par une liste d'unités textuelles. Ces unités sont analysées et donnent, si elles correspondent aux formes du lexique, accès aux informations grammaticales contenues dans le lexique. A partir de ces informations, on construit une représentation du document qui est un enrichissement du document d'origine : le document sera pourvu de métadonnées qui peuvent concerner des unités, mais aussi des parties du document ou bien des parties d'unités.

L'étiquetage morphosyntaxique consiste à fournir une description morphosyntaxique des unités textuelles. Cette description consiste en la catégorie grammaticale et éventuellement des traits morphosyntaxiques. Contrairement à ce qu'on peut penser, nous considérons que le lemme ne fait pas partie de la description morphosyntaxique. La lemmatisation fera l'objet d'un traitement discuté par la suite.

Le terme *morphosyntaxique* mérite une explication. D'un point de vue lexical, la formation des mots relève exclusivement de la morphologie. Si on décrit le paradigme flexionnel d'un mot, on fait une description morphologique du mot : on énumère toutes les formes d'un même mot en appliquant des règles morphologiques à partir d'une forme canonique. Les mots ont des propriétés syntaxiques : en premier lieu la catégorie grammaticale du mot, mais aussi les traits morphologiques relevant de la flexion contextuelle[67]. Ces informations sont utilisées par la désambiguïsation lexicale qui repose sur le contexte syntaxique des mots. Du point de vue que nous prenons, celui de l'étiquetage du mot (lexical, morphologique) ou bien celui de la désambiguïsation (syntaxique, utilisant des connaissances morphologiques), dépend donc notre choix entre les termes respectifs *morphologique* ou *morphosyntaxique*.

Méthodes

La morphologie des langues étant pour la plus grande partie très régulière, même dans ses irrégularités, l'analyse morphosyntaxique peut se faire complètement par règles. Cette méthode est d'autant plus tentante que seule l'approche par règles semble garantir une couverture complète sur les mots communs[68]. On peut mentionner ici la morphologie à deux niveaux introduit par [Koskenniemi, 1983] en alternative viable pour les règles génératives qui étaient le cadre en vigueur depuis les années 60.

La solution la plus simple à implémenter est néanmoins celle du lexique : l'accès aux informations est direct et l'effort de calcul minimal. En revanche, la couverture du lexique est incertaine et un coût de gestion est lié à sa maintenance.

Les méthodes par apprentissage construisent un lexique et des règles ou des probabilités à partir d'un corpus d'entraînement. Si un phénomène n'est pas couvert par ce corpus, l'analyse

---

[67] La distinction entre flexion inhérente et contextuelle a été proposée par [Booij, 1994], séparant les marqueurs imposés par la syntaxe (par exemple les marques d'accord des verbes et des adjectifs) de ceux qui ne le sont pas (par exemple le trait du nombre pour les noms). Il s'en sert pour démontrer que dérivation et flexion ne sont pas deux opérations distinctes, l'une étant présyntaxique (lexical), l'autre post-syntaxique. Ce faisant il réfute l'hypothèse du *split morphology* introduite au début des années 80.

[68] Gare au traitement des noms propres, dans les trois approches, afin de ne pas lemmatiser le nom de ville *Paris* en *pari*…

fournie n'est pas fiable. Pour réduire la dépendance du corpus, un lexique généraliste peut être exploité, ajoutant en contrepartie un effort de gestion non négligeable, vu la quantité de données concernée.

Les systèmes d'analyse de langues, selon qu'ils donnent un poids prépondérant aux règles ou au lexique, peuvent être situés sur un axe, comme dans la figure 15. Sur les extrémités de l'axe nous trouvons d'un coté les systèmes à base de règles sans lexique et les systèmes à base de lexique sans règles de l'autre.

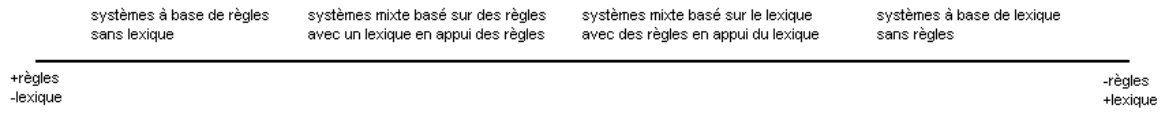


Figure 15 : Axe des systèmes d'analyse morphologique

Lametyzator est un raciniseur polonais qui agit par simple accès au dictionnaire *Morfologik*[69], ce qui explique le grand nombre d'entrées dans ce lexique. Ce raciniseur se trouve donc à l'extrême droite de l'axe ci-dessus. A son opposé, à l'extrême gauche de l'axe, on peut situer *Stempel*, un raciniseur/lemmatiseur heuristique qui n'agit que par des règles apprises sur du lexique. Les deux ont été enchaînés dans le projet Morfologik dans un composant nommé *Stempelyzator*, donnant un léger gain des résultats par la combinaison des deux ([Weiss, 2005]). Notons que les deux composants obtiennent les mêmes résultats séparément seulement si *Stempel* est entraîné sur les formes de 60.000 lemmes, ce qui correspond à un lexique de large couverture. L'approche hybride augmente de quelques pourcents les résultats par lexique, et cela dès un apprentissage sur 5000 lemmes. Comme il contient les connaissances des deux extrémités, on peut placer ce système hybride exactement au milieu, même si dans la pratique le dictionnaire fournit la plupart des analyses grâce à sa couverture.

L'effort de gestion de lexique peut être réduit par l'introduction de règles morphologiques qui agissent conjointement avec le lexique. La quantité de données à gérer est alors fortement réduite avec quelques règles simples d'affixation qui affectent des mots moins fréquents. Par exemple, pour réduire le nombre d'entrées dans le lexique polonais, nous avons introduit une règle en polonais qui interprète l'utilisation du préfixe *nie* (« non ») collé à tout nom, adjectif ou participe présent. Par cette seule règle, nous avons réduit d'un bon tiers le nombre d'entrées du lexique.

PL *niebędącą* = nie + będącą (accusatif/instrumentalis féminin du participe présent de *być*, être) -> lemme : niebędący

nieśmieceniu = nie + śmieceniu (datif/locatif du nom *śmieceni*, le fait de disperser des déchets) -> lemme : nieśmieceniu

nieniemieckojęzycznego = nie + niemieckojęzycznego (adjectif, germanophone) -> lemme : nieniemieckojęzycznego

Le nombre d'entrées économisées allège la gestion du lexique, mais ne donne pas d'estimation de l'utilité de la règle dans l'analyse. Pour la connaître, il faudrait comparer une analyse purement lexicale avec une analyse par règle d'un grand corpus.

[69] Voir http://www.cs.put.poznan.pl/dweiss/xml/projects/lametyzator/index.xml?lang=en et http://morfologik.blogspot.com/.

Les exemples suivants illustrent comment les règles peuvent remplacer des grandes parties de lexique. En espagnol, la pronominalisation des compléments d'objet direct et indirect s'agglutinent à l'impératif, l'infinitif et le gérondif. On voit dans les exemples qu'une des voyelles de la racine peut prendre un accent pendant cette opération. L'intégration de quelques règles permet d'économiser un grand nombre d'entrées par verbe dans le lexique.

ES	*hablarle*	(lui parler)	= hablar + le	
	hablándole	(en lui parlant)	= hablando + le	
	háblale	(parle-lui)	= habla+ le	
	dàmelo	(donne-le-moi)	= da + me + lo	(donne + me + le)

En portugais, un phénomène similaire se produit par l'insertion d'endoclitiques. Au conditionnel et au futur, la pronominalisation du complément d'objet direct se trouve injectée entre le radical du verbe et les terminaisons du verbe, entourée de traits d'union. L'insertion peut en plus introduire des transformations du radical et du pronom, comme dans le deuxième exemple.

PT *inscrever-nos-emos* (nous nous inscrirons)

lavá-las-ão = lavar + as + ão (ils les laveront)

La présence de tirets nécessite néanmoins des règles différentes de celles que nous venons de voir : il ne s'agit plus de décomposer ce qui est collé, mais de composer ce qui est considéré comme des mots séparés. Plusieurs unités textuelles qui se suivent sont rassemblées pour n'en faire qu'une seule.

Pour compléter les règles polonaises, nous pourrions imaginer des règles qui rattrapent des erreurs orthographiques de personnes qui écriraient un espace ou un tiret entre le préfixe *nie* et le mot sur lequel porte la négation.

PL *nie wojskową > niewojskową* (non militaire)

Nous appelons *règles lexicales* les règles qui régissent la composition ou la décomposition d'unités textuelles, et qui, en même temps peuvent fournir une description morphologique sur les unités traitées. Cette grammaire de règles lexicales utilise un lexique de bases morphologiques (aussi appelées *racines*), d'affixes (préfixes, suffixes) et d'endoclitiques qui sont des suites de caractères qui ne sont pas autonomes.

L'optimisation du traitement et de la gestion du lexique en même temps passe par un bon équilibre entre lexique et règles d'analyse morphologique. Les unités fréquentes et irrégulières (ce sont souvent les mêmes) se trouvent dans le lexique, les moins fréquentes et une partie de celles qui sont construites sont interprétées par des règles.

Certains phénomènes de langue se prêtent plus à un traitement par règle que d'autres, comme l'insertion d'endoclitiques ou la préfixation. D'après notre expérience, la suffixation est plus difficile à mettre en œuvre, la préfixation ayant en général peu d'incidence sur les autres caractères du mot ou sur la catégorie de l'unité[70]. Les règles de suffixation sont par conséquent plus complexes. Les exemples suivants de suffixation en néerlandais illustrent cette complexité. Le suffixe *loos* exprime « l'absence » et est un marqueur d'adjectif. L'ajout d'un *e* de liaison a pour effet de doubler la consonne ou la simplification de la voyelle le

[70] Même si c'est possible : en français le préfixe *anti* combiné à un nom le transforme en adjectif, comme dans *une manifestation antimondialisation*.

devançant, conformément aux règles d'orthographe néerlandaise. La base est un substantif, mais l'ensemble un adjectif.

NL	*goddeloos*	= god + e + loos	(impie, littéralement sans dieu)
	tomeloos	= toom + e + loos	(effréné, littéralement sans bride)

Afin de savoir si le codage d'une règle vaut la peine, il faudrait dans l'idéal savoir s'il s'agit d'un phénomène courant ou pas. La productivité est néanmoins impossible à mesurer dans l'absolu : il est possible de chiffrer le nombre d'unités concernées en simulant la règle par une expression régulière sur un corpus, ce qui donne un chiffre relatif. Cependant, comme le nombre de règles est élevé et ce chiffrage peu fiable, on se fie en général à l'intuition du linguiste qui choisit quelles règles implémenter, d'autant plus que le paramètre de l'optimisation du code doit également être pris en compte. Si le système est optimisé pour prendre les informations dans le lexique, alors le phénomène traité par règle ne doit pas être trop courant non plus, sinon le système perd en efficacité.

Pour certaines langues morphologiquement très riches, comme le hongrois, le finnois et le turc, il est impossible de traiter les unités seulement avec du lexique. Dans ces langues, des suffixes, chaînes de caractères sans autonomie qui ne sont pour autant pas des désinences flexionnelles, peuvent s'agglutiner à la fin d'un mot. Ces suffixes sont constructeurs du sens des syntagmes en exprimant ce qui s'exprime dans d'autres langues par des moyens syntaxiques. Pour cette raison, on appelle ces langues agglutinantes, même si le même phénomène existe dans une moindre mesure dans d'autres langues. L'agglutination de l'article défini au nom en suédois est bien une opération d'agglutination, mais on ne considère pas pour autant le suédois comme une langue agglutinante. L'exemple en finnois est une forme fléchie de *talo* (maison), dont le cas est l'inessif[71].

SV	*alv* (elfe), *alven* (l'elfe) = alv + en
	alver (elfes), *alverna* (les elfes) = alver + na
FI	*talossani* = talossa (dans maison) + ni (ma)

Les systèmes qui traitent les langues agglutinées disposent nécessairement d'un composant d'analyse par règles. Pour le hongrois, nous pouvons citer par exemple le système *Humor* ([Prószéky, 1994]) de MorphoLogic qui repose sur un ensemble de bases morphologiques (= lexique) et un ensemble de règles. Pour le finnois, il convient de citer le *Machinese Phrase Tagger* de Connexor dont un bref aperçu peut être trouvé dans [Järvinen et al., 2004][72]. Le modèle du finnois de [Koskenniemi, 1983], qui repose massivement sur des règles, ne convient pas aux systèmes pour lesquels le temps d'exécution de l'analyse est critique. Pour cette raison, le système d'analyse pour le finnois que nous avons mis en place repose fortement sur des règles, mais seulement pour les phénomènes d'agglutination. La flexion fait partie du lexique, même si le système flexionnel est très élaboré, ce qui donne un lexique de taille inégalée. Nous le décrivons un peu plus dans le chapitre 9 qui traite de l'acquisition de lexique.

Les langues agglutinantes sont aussi des langues compositionnelles, c'est-à-dire que la composition de mots composés se fait par agglutination de plusieurs mots autonomes par

[71] L'inessif exprime le lieu dans lequel se déroule l'action exprimée par la phrase.

[72] Tenu compte de la difficulté de traitement des langues agglutinantes, il ne nous semble pas une coïncidence que deux des fournisseurs internationaux de technologies d'analyse de TAL soient hongrois et finlandais. Connexor (http://www.connexor.com/) est fournisseur d'entre autres Lingway (http://www.lingway.com/), Fast (http://www.fastsearch.com/) et Clearforest (http://clearforest.com/).

ailleurs. Si la décomposition en mots autonomes est importante pour l'application, l'utilisation d'un lexique avec les possibles composants et d'une grammaire de décomposition s'imposent. On peut imaginer des approches où le corpus est le lexique et où les règles sont génériques et reposent sur la similarité entre les mots. Le principal problème que nous y voyons est que ce type d'approche pourra décomposer des mots qui ne sont pas à décomposer, comme par exemple *toffee* (caramel) en néerlandais en *tof* (chouette) et *fee* (fée). Il ne saura pas non plus faire le choix de ne pas décomposer. *Federbett* est le mot allemand pour *couette*, et serait décomposé par ce type de système en *Feder* (plume) et *Bett* (lit), ce qui n'est morphologiquement pas faux, mais pas souhaitable pour toutes les applications, notamment pour la recherche d'information. Ou encore la marque de carte de crédit *MasterCard* qui, en anglais, serait décomposé en *Master* (maître) et *Card* (carte).

Le processus de composition étant productif, il faut des règles de décomposition, qui s'appuient sur le lexique, pour donner une description morphologique pour les mots hors vocabulaire. Si l'unité textuelle contient des espaces ou des tirets, la question de l'indexation des composants se pose autant que pour les mots compositionnels, avec seulement la difficulté de la décomposition en moins.

L'équilibre entre règles et lexique a des conséquences directes sur le nombre d'informations lexicales à gérer, et plus particulièrement sur le nombre de mots-formes. Les informations liées aux mots-formes peuvent être plus ou moins riches, augmentant la masse de données à gérer selon le détail de description.

La description grammaticale associée au mot-forme consiste généralement en la catégorie grammaticale et les traits morphologiques. Il existe une certaine hiérarchie entre ces informations. De la catégorie grammaticale dépendent les attributs et certains attributs peuvent en engendrer d'autres. Par exemple, la description d'un nom en allemand aura comme catégorie grammaticale *nom*, les attributs liés au nom sont le genre, le nombre et le cas. Ils auront respectivement une des valeurs *masculin/féminin/neutre*, *singulier/pluriel* et *nominatif/accusatif/datif/génitif*. S'il s'agit d'un mot composé, un attribut optionnel l'indiquera et de cet attribut dépendra l'attribut de la *composition*, qui prendra par exemple la valeur *nom nom* si le mot composé est composé de deux noms. Chaque valeur de catégorie grammaticale possède ainsi un schéma de traits morphologiques qui lui correspond.

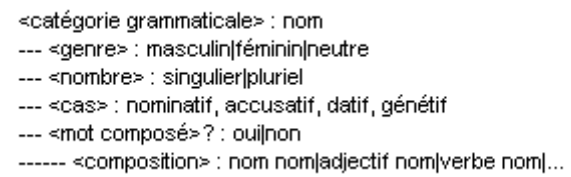


Figure 16 : Schéma du nom en allemand

Ces schémas peuvent être utilisés pour contrôler la cohérence des descriptions fournies dans le lexique, quel que soit le format du lexique. Ces schémas ne sont malheureusement pas universels : ils sont différents pour chaque langue et dépendent de la granularité de description choisie. Ainsi, pour une même langue, le découpage en catégories grammaticales dépend de la source, comme le montrent les quelques comparaisons suivantes. Nous avons pris des mots grammaticaux, car les disparités s'affichent souvent sur ceux-ci.

Mot	Lexique 3.5	Morphalou	Lefff
le	article	functionWord	déterminant
ce	adjectif	functionWord	déterminant
pour	préposition	functionWord	préposition

Tableau 5 : Disparités de catégories grammaticales entre trois lexiques

La différence dans les noms des attributs et des valeurs s'explique en général par une absence de standardisation que cherche à combler l'ISO 24613:2008 avec le répertoire des catégories de données (voir 4.2.5). Parfois, les différences sont plus théoriques, surtout en ce qui concerne les catégories, à moins qu'il ne s'agisse d'erreurs. A titre d'exemple : le mot *oh* est marqué *adverbe* dans Lefff[73], *onomatopée* et *nom* dans Lexique 3.5 et *interjection* dans Morphalou.

Une façon de simplifier la gestion de connaissances d'un système multilingue est d'utiliser les mêmes codes pour différentes langues pour encoder les descriptions. Les propriétés des langues sont limitées, et se ressemblent. Au fur et à mesure que les langues s'ajoutent, le référentiel est complété : des codes sont ajoutés et certains s'interprètent légèrement différemment pour ne pas ajouter inutilement des codes quand un code similaire existe déjà. L'ajout d'un nouveau code demande de connaître tous les codes et donc de disposer d'un accès facile à l'ensemble des codes existants.

Un peu plus haut, nous avions décrit une description morphosyntaxique comme une description grammaticale qui consiste en la catégorie grammaticale et les traits morphologiques. Certains de ces traits sont plutôt sémantiques, comme le marquage de prénom, de nom d'entreprise ou de marque de voiture. [Paroubek et Rajman, 2000] considère « comme pertinente, pour une description morphosyntaxique, toute information permettant de caractériser le comportement d'un mot dans son contexte d'annotation par rapport à l'ensemble fini de traits comportementaux définis *a priori*. » Les noms propres ont effectivement un comportement syntaxique différent des autres noms. Par exemple, quand ils sont utilisés comme nom propre, la majorité de ces noms n'acceptent pas de déterminant devant. Dans la littérature, en dépit de ce comportement différent, ces mots sont tout de même considérés comme des noms propres. Pour la désambiguïsation morphosyntaxique il serait sans doute mieux de disposer d'une catégorie spécifique pour ces noms afin d'exploiter ces différences.

Au regard des opérations morphologiques effectuées, la dérivation et la flexion sont très similaires. L'on pourrait militer pour inclure la dérivation dans les paradigmes flexionnels, mais leur irrégularité mènerait à une multiplication des paradigmes. En contre-argument, nous invoquons également le glissement sémantique qui a lieu : le sens du mot dérivé ne peut pas être confondu au mot d'origine parce que l'affixe a un apport sémantique indéniable.

Connaissances

Après découpage, le fragment est représenté par une liste d'unités textuelles $(u_1, u_2, \ldots, u_n)$. La fonction θ_{etiq} d'analyse et d'étiquetage morphosyntaxique fournit pour chaque unité textuelle u_i, une ou plusieurs descriptions *dM*. Ces descriptions contiennent la catégorie grammaticale *cat* et un ensemble de traits *Tm*. La catégorie et les traits font partie de listes

[73] Lefff : Lexique des formes fléchies du français : http://www.labri.fr/perso/clement/lefff/.

prédéfinies pour chaque langue. Cette fonction peut prendre en compte le contexte pour fournir ces résultats, mais ce n'est pas obligatoire.

$$\theta_{etiq}\left(\left(u_{x-i}, u_x, u_{x+j}\right), C_{etiq}\right) = \left(u, Dm\right)$$

où x = la position de u,

$$Dm = \left\{dM_1, dM_2, ..., dM_n\right\}, n \geq 1$$

avec $$dM = \left(cat_i, Tm_j\right)$$

où $$cat_i \in S,\ S = \left\{cat_1, cat_2, ..., cat_{n'}\right\}$$

et $$Tm_j \subset \left\{tM_1, tM_2, ..., tM_{n''}\right\}$$

Pour ce faire, θ_{etiq} prend en compte un ensemble de connaissances C_{etiq} qui servent à analyser les unités afin d'attribuer une ou plusieurs descriptions à chaque unité.

Deux cas se présentent pour l'analyse des unités textuelles. Dans le premier cas l'unité *u* est présente dans le lexique *Letiq* de la base de connaissances en tant que mot-forme *f* et la mise en correspondance avec l'ensemble des descriptions *Dm* est directe. Aucune analyse n'est faite et on attribue à *u* l'ensemble des descriptions associées à *f*.

Dans l'autre cas, l'unité n'existe pas dans le lexique. Il faut alors passer par une analyse morphosyntaxique en appliquant les règles r_i des grammaires pour la dérivation *Gder*, l'agglutination *Gagg* et la composition *Gcomp*. Ces grammaires sont respectivement constituées de règles *rDer*, *rAgg* et *rComp*. Chaque type de règle essaie de décomposer l'unité textuelle d'une façon différente.

Une règle de dérivation *rDer* décompose une unité textuelle en un mot-forme et un élément. Pour fournir la description Dm_b à l'unité textuelle, la règle prend la description Dm_a du mot-forme en la modifiant éventuellement.

Une règle d'agglutination *rAgg* décompose une unité textuelle en une autre unité textuelle et un élément, en modifiant la description Dm_a qui lui était associée en Dm_b. Cette règle est en effet récursive, ce qui est la propriété discriminatoire de l'agglutination.

Une règle de composition *rComp* décompose une unité textuelle en plusieurs mots-formes et un élément e_i. Elle fournit une description Dm_c à partir des descriptions associées aux mots-formes Dm_a qui lui était associée en Dm_b.

Selon le type de règle appliquée, on peut classer l'unité analysée en mot dérivé, mot agglutiné ou mot composé.

Les éléments e_i qu'utilisent ces règles sont des morphèmes et des séquences de caractères qui ne sont pas autonomes, mais qui peuvent être porteurs de sens. C'est le cas pour les éléments utilisés dans les règles de dérivation (affixes) et d'agglutination (ajouts grammaticaux). Ceux des règles de composition n'ont en revanche pas d'apport sémantique : ils ne servent que de liaison phonétique. Nous regroupons les éléments e_i selon les types des règles qui les utilisent : *Eder*, *Eagg* et *Ecomp*. Chaque élément est utilisé par au moins une règle.

Si aucune règle de ces grammaires n'aboutit, l'unité textuelle analysée est considérée comme hors vocabulaire. Des informations par défaut lui sont associées, c'est-à-dire qu'on obtient une description *d* avec une catégorie grammaticale « inconnu » représentant l'échec d'analyse.

$$C_{etiq} = \{Letiq, Ccomp, Cder, Cagg\}$$

où $Letiq = \{(f_1, Dm_1), (f_2, Dm_2), ..., (f_n, Dm_n)\}$,

$Ccomp = \{Gcomp, Ecomp\}$,

$Cder = \{Gder, Eder\}$,

$Cagg = \{Gagg, Eagg\}$,

$Gcomp = \{rComp_1, rComp_2, ..., rComp_m\}$,

$Gder = \{rDer_1, rDer_2, ..., rDer_{m'}\}$,

$Gagg = \{rAgg_1, rAgg_2, ..., rAgg_{m''}\}$,

$Eder = \{(e_1, Dder_1), (e_2, Dder_2), ..., (e_p, Dder_p)\}$,

$Eagg = \{(e_1, Dagg_1), (e_2, Dagg_2), ..., (e_{p'}, Dagg_{p'})\}$

et $Ecomp = \{(e_1, Dcomp_1), (e_2, Dcomp_2), ..., (e_{p''}, Dcomp_{p''})\}$

où $e_i \in A*$, $A = \{c_1, c_2, ..., c_n\}$,

$Dder = \{dDer_1, dDer_2, ..., dDer_q\}$,

$Dagg = \{dAgg_1, dAgg_2, ..., dAgg_{q'}\}$,

$Dcomp = \{dComp_1, dComp_2, ..., dComp_{q''}\}$,

$\forall (e_i, Dder_i) \in Eder, \exists rDer_j : rDer_j(u_j, e_i) = ((f_k, Dm_a), e_i), (u_j, Dm_b)$,

$\forall (e_i, Dagg) \in Eagg, \exists rAgg_i : rAgg_i(u_j, e_i) = ((u_k, Dm_a), e_i), (u_j, Dm_b)$,

et $\forall (e_i, Dcomp) \in Ecomp \; \exists rComp_i : rComp_i(u_j, e_i) = ((f_a, Dm_a), e_i, (f_b, Dm_b)), (u_j, Dm_c)$

où $dDer = (catDer, Tder)$, $catDer \in S$, $Tder \subset T$

$dAgg = (catAgg, Tagg)$, $catAgg \in S$, $Sagg \subset S$, $Tagg \subset T$

et $dComp = (catComp, Tcomp)$, $catComp \in Scomp$, $Scomp \subset S$, $Tcomp \subset T$

5.5 Désambiguïsation morphosyntaxique

Objectifs

Hors contexte, certaines unités textuelles sont naturellement ambiguës grammaticalement (par exemple *avions* qui est *nom* et *verbe*). Le traitement de la désambiguïsation morphosyntaxique vise à réduire cette ambiguïté en prenant en compte le contexte grammatical de chaque unité textuelle.

Méthodes

Toutes les approches de désambiguïsation morphosyntaxique ne se retrouveront pas dans notre description très générale. Depuis les années 90, différentes méthodes ont été élaborées. [Paroubek et Rajman, 2000] les classifient en *méthodes à base de règles sans apprentissage, méthodes à base de règles avec apprentissage, méthodes probabilistes, méthodes à base de*

réseaux de neurones artificiels, et *autres*. Les premières méthodes, à base de règles, sont souvent qualifiées de *linguistiques*, pour la raison que les connaissances mises en œuvre sont quelques centaines voire milliers de règles qui sont l'œuvre d'experts linguistes. [Karlsson et al., 1994] est un exemple parfait d'un tel système, implémentant environ 1100 règles pour désambiguïser l'anglais, obtenant un taux d'erreur initial de 0,3%. L'adaptation spécifique à un des trois sous-corpus, l'encyclopédie Grolier, faisait tomber le taux d'erreur à 0,04%. La méthode la plus célèbre se classe néanmoins dans les règles apprises sur corpus : il s'agit des travaux de [Brill, 1992] et [Brill, 1995]. Toutes les méthodes ont en commun de s'appuyer sur le contexte du mot à désambiguïser pour déterminer sa catégorie.

La méthode de Brill et les méthodes probabilistes ont en commun qu'elles ont besoin d'un corpus d'apprentissage pour créer ce que nous appelons un *modèle de langage*[74], l'ensemble des connaissances utilisées pour la désambiguïsation. Dans le cas du système de Brill, il s'agit principalement d'un lexique et d'une liste de règles. C'est selon cette méthode que nous avons formalisé le traitement de la désambiguïsation ci-dessus.

D'après [Jelinek, 2004] c'est Bob Mercer chez IBM qui aurait dit en 1985 « There's no data like more data »[75]. [Banko et Brill, 2001] démontrent qu'il vaut mieux étendre le corpus d'apprentissage plutôt que de complexifier l'algorithme. C'était aussi l'avis du conférencier invité à TALN 2007, Dekang Lin de Google Inc., qui obtient de très bons résultats d'analyse avec des méthodes statistiques superficielles grâce à la grande quantité de données dont il dispose. Kenneth Church, dans sa conférence invitée à TSD 2004 [Church, 2004], prédit un retour au « rationalisme », après des années d'empirisme, à partir de 2010 : retour aux méthodes linguistiques plus théoriques plutôt qu'une approche uniquement basée sur les données. [Cieri et Liberman, 2008] rapporte en effet que la demande n'est plus dans la quantité de données, mais qu'elle se situe dans la qualité, c'est-à-dire une annotation plus fine et précise des données. La quantité de données produites et livrées dans certains programmes d'évaluation est telle que les systèmes ne peuvent pas s'entraîner sur les corpus complets dans les temps impartis.

Les méthodes statistiques de désambiguïsation morphosyntaxique ont fait leurs preuves : dans l'état de l'art, les chiffres de 95% à 99% sont courants. Pour calculer le nombre de phrases correctement étiquetées, on calcule le taux approché d'étiquetage correct des phrases avec la formule suivante qui est mentionnée dans [Paroubek et Rajman, 2000]: $(t_{corr})^n$ avec t_{corr} étant le taux d'étiquetage correct et *n* le nombre moyen de mots par phrase. Dans ses travaux sur la maturité syntaxique, [Gagnon, 1998] avait constitué et analysé un corpus pour chiffrer la moyenne des mots par phrase à l'âge de 18 ans pour le comparer avec celle des journalistes. Le nombre moyen de mots par phrase était respectivement de 17,87 et de 25,2. Il est évident que ce chiffre varie fortement selon le type de corpus (et selon la langue), et que même basé sur corpus, toute moyenne est relative. Pour cette raison, nous donnons les valeurs pour des phrases moyennes de 15, 20 et 25 mots. Les valeurs que nous obtenons en appliquant la formule ci-dessus expriment le pourcentage de phrases bien étiquetées.

[74] Ce terme est plus usuel pour désigner un modèle sans règles linguistiques, comme pour les méthodes probabilistes ou en transcription automatique.

[75] A interpréter dans son contexte. En dans les années 80 les méthodes de transcription automatique de la parole connaissent de grands progrès en délaissant les méthodes linguistiques pour des méthodes d'apprentissage.

Précision	15 mots	20 mots	25 mots
0,95	46	36	28
0,96	54	44	36
0,97	63	54	47
0,98	74	67	60
0,99	86	82	78

Tableau 6 : Taux approché d'étiquetage

Si on prend la moyenne arbitraire de 20 mots, à 0,95 de précision, on obtient seulement 36 % de phrases correctement étiquetées, ce qui veut dire que seule une phrase sur trois ne contient aucune erreur d'étiquetage. A 0,98 de précision, on obtient 67 % de phrases bien étiquetées, soit deux phrases sur trois.

La plupart des expériences sont néanmoins faites sur l'anglais, souvent faute de ressources, et la transposition sur d'autres langues n'est pas toujours garantie pour la raison que les contraintes syntaxiques et le taux d'ambiguïté par mot sont variables selon les langues. Comme le montre [Hajič, 2000] dans une expérience sur cinq langues de l'Europe de l'est, une plus grande richesse morphologique demande un corpus d'apprentissage bien plus grand. Il vaut donc mieux investir dans l'analyse lexicale pour ces langues.

L'interprétation des chiffres mentionnés ci-dessus n'est pas triviale. La comparaison de plusieurs systèmes demande qu'ils se soient par exemple accordés sur le même jeu d'étiquettes, un jeu plus réduit pouvant constituer un avantage considérable. La comparaison entre systèmes de différentes langues est alors encore plus dure sachant que le nombre d'étiquettes dépend des caractéristiques de la langue : [Hajič, 2000] mentionne, en se fondant sur Multext-East, des jeux de près de 500 étiquettes pour l'estonien et le hongrois, de près de 1000 étiquettes pour le slovène et le tchèque, tandis que l'anglais en a 139. Des expériences effectuées à Sinequa ont montré que plus nombreux sont les traits pris en compte pour l'apprentissage, plus grand doit être le corpus pour donner des résultats acceptables à l'apprentissage.

En plus, comme le pointe justement [Kilgarriff, 2003], les conditions rencontrées sur le terrain ne sont pas celles du laboratoire. Le protocole d'évaluation de la performance d'un étiqueteur veut que les corpus d'entraînement et de test soient du même type afin de pouvoir pleinement évaluer les algorithmes implémentés. Si on change de domaine, les performances chutent inévitablement. Pour obtenir de bons résultats, le corpus d'entraînement doit donc ressembler au corpus qui sera traité par l'application finale.

Pour chaque application qui demande l'inclusion d'un module de désambiguïsation morphosyntaxique se pose alors la question du type de modèle à utiliser. Si les corpus à traiter sont homogènes, d'un domaine spécifique, alors il vaut mieux essayer d'adapter le modèle utilisé par le lourd processus d'élaboration d'un corpus d'apprentissage. Si le corpus est hétérogène ou inconnu, ou s'il évolue dans le temps, se pose la question comment y faire face sans intervention humaine explicite. Le choix se pose entre avoir plusieurs modèles à disposition pour utiliser celui qui est le plus adapté, et reposer sur un seul modèle neutre aux domaines ou généraliste. Dans ce dernier cas se pose ensuite le problème de la création du modèle. Un modèle généraliste, déployable sur tout type de corpus, suppose d'élaborer un corpus généraliste.

A notre connaissance, il n'existe aucune méthode pour constituer un tel corpus, la notion de « corpus généraliste » allant même à l'encontre de la définition de corpus. En effet, celui-ci

est par définition homogène. Nous proposons de considérer un corpus hétérogène comme un ensemble de corpus homogènes qui respecte une proportion de diversité prédéfinie. Concernant la constitution, cela déplace le problème de l'hétérogénéité au choix des sujets des sous-corpus homogènes. Le BNC est un exemple d'un corpus qui se voulait représentatif de la langue, aussi bien en langue parlée qu'en langue écrite, et tous thèmes confondus. [Burnard, 2000], qui offre un regard rétrospectif sur l'histoire du BNC, parle d'un repositionnement rapide vers un corpus fait de sous-corpus spécialisés, avant de conclure sur un manque de clarté dans la taxonomie des types de textes et de militer pour un meilleur accès à ceux-ci.

L'inclusion de certains types de textes doit néanmoins être évitée à tout prix, car ils risquent de casser le modèle de langage. Il s'agit de tout texte dont le contenu ne correspond pas à des suites syntaxiques, tels que des tableaux de tableur contenant par exemple les coordonnées d'un ensemble de personnes sur plusieurs colonnes.

Une autre stratégie pour rendre la désambiguïsation moins dépendante du corpus d'entraînement est de brancher un lexique général, encore appelé de large couverture. Ce type de lexique contient idéalement tous les mots de la langue et ajoute des informations qui sont absentes du modèle de langage pour la simple raison que les mots en question n'étaient pas présents dans le corpus d'apprentissage. En effet, les étiqueteurs qui ont appris leur modèle de langage sur un corpus construisent un ensemble cohérent de lexique et de règles pour étiqueter et désambiguïser du texte. Cela veut néanmoins dire qu'on ne peut pas toucher à ces informations à moins de rendre le modèle de désambiguïsation caduc. La seule façon de corriger un tel modèle est d'intervenir sur le corpus et de réapprendre le modèle. L'ajout d'un lexique général rend un tel étiqueteur plus robuste dans le sens où on limite les mots « hors vocabulaire »[76]. Cela peut aussi être obtenu en branchant un devin[77], qui lui aussi peut être fait à partir de plusieurs méthodes. Si le choix du lexique est fait, il faut le créer, le gérer et savoir faire la fusion des connaissances.

L'ajout d'un traitement de désambiguïsation entraîne des contraintes sur les connaissances du traitement précédent, l'analyse et l'étiquetage, dont l'origine se situe dans la difficile cohabitation de théorie linguistique et traitement de TAL. La nature des mots est un de ces problèmes. La catégorie grammaticale de *nature* dans *des nouilles nature* est un nom qui se trouve en position adjectivale. Si on l'étiquette nom, on admet la suite *nom nom* dans le modèle de langage, ce qui n'est pas courant du tout. Si on l'étiquette comme *adjectif* dans le corpus d'entraînement, le lexique le contiendra aussi comme un adjectif, ce qui ne semble pas correct. En français, on peut mentionner la distinction entre nom et adjectif pour un très grand nombre de mots comme par exemple *acadien*, *alcoolique*, *ami*, *adhésif* ou *bonapartiste*. Coder tous ces mots comme nom et comme adjectif crée des ambiguïtés inutilement. La distinction entre participe passée et adjectif est un problème du même type. Si pour certaines entrées la distinction semble claire (par exemple *parfait* ou *décédé*), ce n'est pas le cas pour d'autres. Pour chaque participe passé, il faudrait examiner si le sens utilisé en position adjectivale est différent de celui du verbe, afin de ne pas le lemmatiser comme verbe dans l'étape suivante. L'étiquetage du corpus doit être cohérent avec le lexique et ne pas simplement réfléchir des positions syntaxiques. En russe et en polonais, le même problème existe pour la distinction entre participe présent et adjectif. Des problèmes similaires existent dans les autres langues : à chaque fois il faut d'abord vérifier s'il s'agit d'une propriété syntaxique de la catégorie (par exemple en néerlandais tout infinitif peut se trouver en position nominale) et si le sens du mot est altéré. A partir de là, on décide s'il faut dupliquer

[76] De l'anglais : OOV ou *out of vocabulary*.
[77] *Guesser* en anglais.

les descriptions ou pas. Evidemment, il faut avoir résolu ces problèmes qui semblent exclusivement lexicaux avant d'étiqueter le corpus d'apprentissage, sinon les résultats ne correspondront pas aux attentes. Et si un lexique général est ajouté au traitement, il faut veiller à ce que lexique et corpus ne soient pas contradictoires.

Quand on fait de l'annotation manuelle de corpus en vue de créer un modèle par entraînement, on se heurte aussi à des problèmes linguistiques insoupçonnés. Nous n'avons par exemple pas trouvé de manuel décrivant comment étiqueter les chiffres des énumérations (1.1, 1.2 etc.), des chiffres romains ou des chiffres écrits. Nous avons considéré ces phrases comme des anomalies (ce qu'elles ne sont pas en réalité) qui pourraient casser le modèle de langage appris à partir du corpus. Nous avons aussi en vain cherché une catégorie adéquate pour des mots comme *bonjour*, *oui*, *d'accord* quand nous avons corrigé l'étiquetage d'un corpus de transcriptions de conversations. Ce ne sont pas des mots sémantiquement vides comme les interjections (*hmm*) mais des éléments pragmatiquement importants et indispensables à la conversation. Dans ce sens il est difficile de les traiter comme des adverbes, d'autant plus qu'ils peuvent constituer une phrase, par exemple *Bonsoir Monsieur*. Nous avons estimé que ces éléments disposent de propriétés syntaxiques spécifiques qui justifient qu'on les attribue une nouvelle catégorie d'*élément pragmatique*.

Connaissances

Le texte annoté $frag_{etiq}$ en sortie de la fonction d'analyse et d'étiquetage d'unités textuelles $\theta_{etiq}(frag)$ est une liste d'unités textuelles avec leurs descriptions. Une unité peut en effet avoir plusieurs descriptions en raison de l'ambiguïté morphosyntaxique des unités.

$$frag_{etiq} = ((u_1, D_1), (u_2, D_2), ..., (u_m, D_n))$$

où $D_i = \{d_1, d_2, ..., d_o\}$

et $m \geq n$, $o \geq 1$

La désambiguïsation morphosyntaxique θ_{des} consiste à réduire le nombre de descriptions associées à une même unité textuelle, la situation idéale étant une seule description correcte par unité. Pour cela elle exploite les connaissances C_{des} qui contiennent un ensemble de modèles de langage *mdl*. Un modèle de langage consiste en un lexique de mots-formes *f* et en une grammaire de désambiguïsation *Gdes*. A tout modèle de langage correspond une liste *cor* de couples d'unités et de descriptions *D*. Il est généré par le traitement d'apprentissage θ_{app}.

$$\theta_{des}\left(\left(u_{x-1}, D_i\right), \left(u_x, D_j\right), \left(u_{x+1}, D_k\right), mdl_{ld}\right) = \left(\left(u_{x-1}, D_m\right), \left(u_x, D_n\right), \left(u_{x+1}, D_o\right)\right)$$

où x est une position

$D_m \subset D_i$,

$D_n \subset D_j$,

$D_o \subset D_k$

et $mdl \in C_{des}$

$$C_{des} = \left\{mdl_1, mdl_2, ..., mdl_n\right\}$$

où $mdl = \left(L, Gdes\right)$

où $L = \left\{f_1, f_2, ..., f_n\right\}$

et $Gdes = \left\{rDes_1, rDes_2, ..., rDes_n\right\}$

$$\forall mdl \; \exists cor : \theta_{app}\left(cor\right) = mdl$$

où $cor = \left(\left(u_1, d_1\right), \left(u_2, d_2\right), ..., \left(u_n, d_n\right)\right)$

5.6 Lemmatisation et racinisation

Objectifs

La lemmatisation et la racinisation[78] sont des traitements qui sont couramment utilisés en RI pour indexer un certain nombre d'unités textuelles sur la même clé. Dans l'index, les unités textuelles y sont représentées par une séquence de caractères qu'on appelle lemme ou racine. La lemmatisation réduit essentiellement les formes produites par le phénomène linguistique de la flexion, alors que la racinisation couvre en plus le phénomène de la dérivation. Lemmes et racines sont utilisés comme clés d'indexation représentant un certain nombre d'unités textuelles, ce qui permet de réduire la taille de l'index si on renonce à la possibilité de la recherche exacte, mais surtout d'augmenter la précision et le rappel dans un processus de recherche.

Méthodes

La racinisation est l'opération associant une racine à une unité textuelle. Si la racinisation est basée sur des règles – au moins dans une approche comme celle de [Fox et Fox, 2002] qui sépare la grammaire du programme – la seule ressource du traitement est la grammaire de racinisation G_{rac}.

Racine ne doit pas être interprétée au sens linguistique du mot. Il s'agit d'une séquence de caractères identifiée par des règles comme racine et qu'un ensemble d'unités textuelles ont en commun. La racine de *porter*, *Porter*, *portage*, *port* et *porteur*, *portier*, *portable* serait *port*. La racine est en général calculée par règle, mais rien n'empêche de la pré-calculer pour des unités connues et de la fournir par lexique. Une caractéristique de la racinisation est qu'elle ne peut s'appliquer que sur des unités textuelles simples. L'exemple donné montre que la

[78] Mieux connu sous le terme anglais *stemming*. Nous traduirons *stemmer* par *raciniseur*.

racinisation établit des liens sémantiques au-delà des liens morphologiques, mélange de niveaux de traitement qui cause des problèmes. Ainsi le *port* (de Cherbourg) et le *port* (de la ceinture) se trouvent associés au *portage* (salarial), au *porteur* (de lumière) et au (raciniseur) *Porter*, comme s'il s'agissait d'une seule et même famille de mots.

La lemmatisation est l'opération consistant à fournir un lemme pour une unité textuelle. Si plusieurs descriptions sont fournies, le mot est ambigu et la lemmatisation livre autant de lemmes que de descriptions. Le lemme est une forme choisie par convention dans les formes du paradigme flexionnel de l'unité concernée.

Si la lemmatisation repose en général sur des lexiques et des grammaires, il est possible de procéder à une lemmatisation sans ressource lexicale. [Koskenniemi, 1983] décrit dans les travaux antérieurs qu'il mentionne, notamment sur le suédois, mais aussi sur le français et l'allemand, des approches sans lexique pour regrouper les formes fléchies d'un même mot, avec une précision du regroupement et de « lemmatisation » en suédois de respectivement 95 et 85 %. Nous considérons que ces résultats sont insuffisants pour notre application.

[Moulinier et al., 2001] comparent l'effet en français d'une racinisation de type [Porter, 1980] avec une approche de lemmatisation dictionnairique. Le premier s'avère parfois trop agressif, ramenant par exemple *français* à *franc*. Ils indiquent néanmoins à juste titre que la qualité du lemmatiseur dépend de la qualité de son lexique, et que les résultats du raciniseur étaient meilleurs si les mots étaient hors vocabulaire. La lemmatisation utilisée dans l'expérience est complètement statique, et aucune règle n'est prévue pour traiter les mots hors vocabulaire. L'utilisation d'un dictionnaire ou pas est un faux argument dans cette discussion. Un lemmatiseur peut être complètement ou partiellement bâti avec des règles, ce qui lui évite les inconvénients de la couverture. La question sous-jacente est plutôt de savoir si dans un système de RI on indexe des lemmes ou des racines, sachant qu'ils ne couvrent pas la même chose. La lemmatisation couvre le phénomène de la flexion et a l'avantage de la précision, la racinisation couvre la flexion et la dérivation, ayant l'inconvénient de l'imprécision (comme par exemple *couper* qui mène vers *coupable*). Le système théoriquement le plus précis est une lemmatisation assortie d'un module qui gère la dérivation. Selon le système qu'on souhaite construire, il faut faire ses choix. Un système combinant la précision de la lemmatisation avec une liste de dérivations semble donc la meilleure solution, mais aussi la plus coûteuse à mettre en œuvre.

La lemmatisation à partir de lexique repose sur une large couverture des unités textuelles. Pour cela, le lexique est constitué de *paradigmes flexionnels*, un ensemble de couples (mot-forme, description) dans lequel toutes les formes ont la même racine linguistique mais des flexions différentes, la même catégorie grammaticale et la ou les mêmes significations. L'absence de flexion compte aussi pour une flexion.

Le paradigme flexionnel *Pf* du nom français *manœuvres* contient sa forme au singulier et au pluriel. L'exemple ci-dessous montre une description minimale, consistant en la seule catégorie grammaticale associée à l'unité textuelle. Cette description est minimale et obligatoire, car la catégorie est nécessaire pour déterminer le paradigme quand l'unité est ambigüe. Elle fait la différence pour le mot *manœuvres* entre le nom et le verbe. Un couple (*unité, description*) donne comme paradigme flexionnel un ensemble de couples (*mot-forme, description*).

```
FR      Pf (manœuvres, nom) = {
                (manœuvre, nom),
                (manœuvres, nom)
                }
```

FR Pf (manœuvres, verbe) = {
(manœuvre, verbe),
(manœuvres, verbe),
(manœuvrons, verbe),
(manœuvrez, verbe),
(manœuvrent, verbe),
…
}

Pour un nombre limité de mots, l'information donnée par la catégorie ne suffit pas. En français, le genre distingue par exemple *le* mousse de *la* mousse[79]. Il faut donc fournir plus d'informations, en l'occurrence le genre, pour effectuer une analyse plus précise. Les paradigmes de flexion sont néanmoins les mêmes pour les deux mots. Dans ce cas il est impossible de différencier les deux paradigmes seulement par le lemme, puisqu'il est le même.

FR Pf (mousse, nom masculin) = {
(mousse, nom masculin singulier),
(mousses, nom masculin pluriel)
}

FR Pf (mousse, nom féminin) = {
(mousse, nom féminin singulier),
(mousses, nom féminin pluriel)
}

Morphalou résout ce problème en faisant la différence entre entrées et lemmes, en ajoutant un chiffre au lemme pour donner un identifiant à l'entrée. Le lemme des deux noms est bien *mousse*, mais les entrées sont identifiées par *mousse_1* et *mousse_2*. La différence entre catégories y est réglée de la même façon : il existe un *mousse_3* qui est un adjectif, dont le lemme est également *mousse*[80].

[Ferdeghini et Niggi, 2001], paragraphe 303, recensent une dizaine de mots qui ont la même forme au singulier, mais qui ont plusieurs formes au pluriel. En voici un exemple en italien.

Lemme	description	mots-formes	traduction
ciglio	*ciglio*	nom masculin singulier	bord, cil
ciglio	*cigli*	nom masculin pluriel	bord (ex. les bords d'une route)
ciglio	*ciglia*	nom féminin pluriel	cil (ex. les cils humains)

Remarquons au passage que *ciglia* au pluriel a changé de genre par rapport à son singulier, phénomène que la même grammaire illustre avec une autre dizaine d'exemples. *Ciglio*/*cigli* et *ciglio*/*ciglia* ont tous le même lemme : *ciglio*.

Plusieurs mots avec des significations différentes peuvent donc avoir le même lemme, ce qui rend le lemme inutilisable comme unité de base pour des applications qui nécessitent une sémantique précise, même si les cas semblent plutôt rares.

Nous avons rencontré un phénomène semblable en néerlandais, sans pour autant pouvoir quantifier son ampleur. En voilà un exemple :

---

[79] *Une* mousse : classe de plantes ; *un* mousse : jeune garçon sur un navire apprenant le métier de marin.
[80] Le Petit Robert : Qui n'est pas aigu ou qui n'est pas tranchant (ex. *pointe devenue mousse par usure*)

Lemme	description	mots-formes	traduction
werk	werk	nom singulier	œuvre, travail
werk	werken	nom pluriel	œuvres, travaux
werken	werken	nom pluriel	travaux publics
werken	werken	verbe infinitif	travailler

Au niveau morphosyntaxique, il est impossible de choisir le lemme pour le nom *werken* : *werk* ou *werken*. Ce type d'ambiguïté nous semble très courant en néerlandais avec un léger glissement de sens, comme pour les mots suivants qui expriment, toujours en néerlandais, une *matière* au neutre, une *pièce de cette matière* au masculin.

Pf (diamant, nom neutre) = {
(diamant, nom neutre singulier)
}
	diamant (matière)

Pf (diamant, nom masculin) = {
(diamant, nom masculin singulier)
(diamant, nom masculin pluriel)
}
	diamant (pièce)

Pf (katoog, nom neutre) = {
(katoog, nom neutre singulier)
}
	œil-de-chat (matière)

Pf (katoog, nom masculin) = {
(katoog, nom masculin singulier)
(katoog, nom masculin pluriel)
}
	œil-de-chat (pièce)

Au vu de ces exemples italiens et néerlandais, le couple <*lemme, catégorie grammaticale*> ne suffit pas pour caractériser une entrée de lexique. Il faut y ajouter une référence à une *classe de flexion* pour distinguer ces mots qui ont le même lemme, mais pas les mêmes mots-formes dans leur paradigme. La classe de flexion est la généralisation des paradigmes de flexion.

Du point de vue lexical, toutes les formes d'un même paradigme de flexion sont des produits du lemme du paradigme et de sa *classe de flexion*. Cette classe est l'ensemble des opérations minimales à exécuter pour obtenir à partir du lemme toutes les formes du paradigme et les descriptions associées. La classe de flexion se définissant à partir du lemme, les règles pour le choix du lemme dans le paradigme de flexion doivent être bien définies. Le choix du lemme dans le paradigme se fait par convention et par analogie à ce qui se fait en lexicographie pour les dictionnaires papier.

La classe de flexion est en général représentée par un code. L'application d'une classe de flexion à un mot donne le paradigme de flexion de ce mot, c'est-à-dire tous les couples <*mot-forme*, *description*> associés au mot en question. L'exemple ci-dessous illustre la classe de flexion du mot allemand *Axiom*. Les opérations marquées sont faites à partir du lemme classique, le nom au nominatif singulier. Sous cet angle de vue, le lexique se présente plutôt sous forme d'arbre, avec aux feuilles les formes.

nom	- neutre	- nominatif	- singulier	ø	Axiom
			- pluriel	+e	Axiome
		- accusatif	- singulier	ø	Axiom
			- pluriel	+e	Axiome
		- datif	- singulier	ø	Axiom
			- pluriel	+en	Axiomen
		- génitif	- singulier	+s/+es	Axioms/Axiomes
			- pluriel	+e	Axiome

Deux lemmes ont la même classe de flexion si les opérations pour obtenir leur paradigme de flexion sont exactement les mêmes. Par exemple, *certitude* et *démonstration* appartiennent à la même classe de flexion car en appliquant les mêmes règles d'une classe de flexion, on obtient leurs paradigmes flexionnels complets. La classe de flexion est la suivante :

nom féminin	- singulier	ø
	- pluriel	+s

Son application aux lemmes *certitude* et *démonstration* donne respectivement :

Pf (certitude) = {(certitude, singulier), (certitudes, pluriel)}
Pf (démonstration) = {(démonstration, singulier), (démonstrations, pluriel)}

En allemand, *Beispiel* (exemple), *Bier* (bière), et *Diktat* (diktat) sont quelques mots qui ont la même classe de flexion que *Axiom*. Si on représente cette classe par le code N1, l'on peut représenter les informations lexicales par un ensemble de couples (lemme, classe de flexion).

L = {(Axiom,N1), (Beispiel,N1), (Bier,N1), (Diktat, N1), …}

La factorisation n'est complète que si l'on regroupe les lemmes par classe de flexion:

L = {(N1,{Axiom, Beispiel, Bier, Diktat,…}),(N2,{…}),…}

C'est donc le triplet <lemme, catégorie grammaticale, classe de flexion> qui identifie une entrée morphosyntaxique de lemme dans la base de connaissances. Si l'analyse morphosyntaxique fait la distinction entre ces mots, elle doit prendre en compte les traits morphologiques dans son modèle de langage. Dans ce cas, le lemme tout seul n'est pas une bonne clé pour l'indexation, car il ne différencie pas la classe de flexion.

Lemmes et racines ne sont donc théoriquement pas de bonnes clés pour une indexation, les racines pour leur imprécision, les lemmes pour leur ambiguïté sémantique. La littérature scientifique semble dire en même temps le contraire, comme nous l'avions vu plus haut, considérant que la recherche d'information ne requiert pas une telle finesse dans les traitements.

A l'origine des erreurs de lemmatisation on trouve en général une mauvaise ou une absence de désambiguïsation morphosyntaxique. L'exemple suivant en portugais en est un bon exemple.

Lemme	description	mots-formes	traduction
ponteira	nom féminin	*ponteira, ponteiras*	frette, bout ferré
ponteiro	nom masculin	*ponteiro, ponteiros*	aiguille, baguette, médiator

Ponteiro et *ponteira* sont deux noms qui n'ont aucune forme en commun, et la lemmatisation se fait en formes distinctes. Il existe néanmoins également un adjectif dont le lemme est *ponteiro*. Cet adjectif rassemble toutes les formes des noms *ponteiro* et *ponteira*. Si la désambiguïsation ne se fait pas ou mal, alors tous ces mots sont indexés sur le même lemme, et une recherche sur le nom *ponteiros* peut donner le nom *ponteiras*.

Lemme	description	mots-formes	traduction
ponteiro	adjectif	*ponteiro, ponteira, ponteiros, ponteiras*	de pointe, d'avant

On a le même cas en portugais pour les noms *meio* et *meia* d'une part, et l'adjectif *meio* de l'autre.

Lemme	description	mots-formes	traduction
meio	nom masculin	*meio, meios*	milieu, média, canal, chaîne
meia	nom féminin	*meia, meias*	bas, chette
meio	adjectif	*meio, meia, meios, meias*	demi-

Il existe un nombre restreint de mots pour lesquels la lemmatisation est problématique : les mots contractés comme les exemples suivants en français et en polonais.

FR *au = à le*
aux = à les
du = de le
des = de les

PL *nań = na niego* (sur lui)
doń = do niego (à/pour lui)
przezeń = przez niego (par lui)

Le phénomène phonétique à l'origine de ces contractions se retrouve exprimé à l'écrit, et crée ce qu'on pourrait appeler des "anomalies structurelles" dans la syntaxe si on considérait les mots contractés comme des unités indivisibles. Pour la création du modèle de langage, on peut les considérer comme unités indivisibles auquel cas ils seront des traits forts dans le modèle, ou bien les décomposer comme des suites de différents mots auquel cas ils renforceront les suites des catégories existantes. Ce choix détermine la façon de coder les mots contractés dans le lexique. On constate ainsi des disparités de codage entre lexiques :

	Lexique3	Morphalou 2.0
du	du	de / le
des	des	de / les
au	au	à / le
aux	aux	à / les

En effet, Lexique3 lemmatise les formes par elles-mêmes, tandis que Morphalou les recense comme formes de chaque entrée des composants. Les mots contractés étant des mots grammaticaux, ils ne sont pas d'une importance directe en recherche d'information, mais ils sont néanmoins très importants pour la compréhension du texte.

Ajoutons pour la complétude que, dans certaines langues, les noms propres aussi disposent de flexions, et sont donc sujets à lemmatisation. On peut mentionner le suffixe *s* en néerlandais et en allemand, marque du pluriel ou du possessif, mais le phénomène est plus impressionnant dans les langues slaves et finno-ougriennes, où les prénoms, les patronymes s'il y en a et les noms de familles se déclinent comme si c'étaient des noms communs. Les noms propres étant une classe ouverte, leurs formes ne peuvent être recensées exhaustivement et leur détection et lemmatisation se font forcément par des règles. [Piskorski et al., 2007] illustre bien la difficulté de la lemmatisation des noms de personne en polonais. Nous avons pu travailler sur les noms de personne en russe qui présentent exactement le même problème, les patronymes en plus. L'exemple russe montre la transcription de Barack Obama au nominatif dans le 1^{e} exemple, au génitif dans le second. Sa déclinaison est intéressante : comme il ne s'agit pas

d'un nom russe, les règles appliquées pour le nom de famille ne se puisent pas dans les classes de flexion habituelles des noms de famille, mais dans les noms communs. Comme le nom de famille se termine par *a*, le mot est décliné comme un nom féminin, la flexion du prénom restant masculin.

NL	Frederik**s** jury	le jury de Frederik
	Frederik Cailliau**'s** jury	le jury de Frederik Cailliau
RU	На фото: 44-й президент США **Барак Обама** во время своей первой пресс-конференции в Чикаго. (Photo: 44e président américain **Barack Obama**, lors de sa première conférence de presse à Chicago.)	
RU	Темнокожие первопроходцы от Абрама Петровича Ганнибала до **Барака Обамы** (Pionniers noirs de Abraham Petrovitch Hanibal à **Barack Obama**.)	

Le rôle que peuvent jouer les paradigmes de flexion dans la gestion des ressources sera discuté en 9.1 (p. 202) avec le finnois en exemple.

Connaissances

Racinisation :

$$\theta_r(u, G_r) = rac$$

où $\forall\, u \in U, \forall\, rac \in R, \#rac \leq \#\mathrm{u}$

$$longueur(rac) \leq longueur(u)$$

et $G_r = \{r_1, r_2, ..., r_n\}$

Lemmatisation :

$$\theta_{lemm}((u, D), C_{lemm}) = (lem_1, lem_2, ..., lem_n)$$

où $D = \{d_1, d_2, ..., d_m\}$

et $lem_i \in P_f(u, d_i)$

Les connaissances C_{lemm} regroupent donc des règles et du lexique. Le lexique contient minimalement des couples (forme, lemme), mais les triplets (forme, lemme, description) apportent plus de précision.

$$C_{lemm} = \{Glemm, Llemm\}$$

où $Glemm = \{rLemm_1, rLemm_2, ..., rLemm_n\}$

et $Llemm = \{(f_1, lem_1, D_1), (f_2, lem_2, D_2), ..., (f_m, lem_m, D_m)\}$

où $(f_i, lem_i) \neq (f_j, lem_j)$

Si le lexique est organisé par des combinaisons de lemmes et de classes de flexion *cf*, le lexique peut être représenté de la façon suivante :

$$Llemm = \{(lem_1, cf_1)_1, (lem_2, cf_2)_2, ..., (lem_m, cf_m)_n\}$$

où $(lem_i, cf_i) \neq (lem_j, cf_j)$

et $\forall\, cf\ \exists\, Gflex : \theta_{flex}(Gflex, (lem, cf)) = pf(lem)$

où $pf(lem) = \{(f, lem, D), (f, lem, D), ..., (f, lem, D)\}$

et $Gflex = \{rFlex_1, rFlex_2, ..., rFlex_n\}$

5.7 Renvois entre unités textuelles

Objectifs

Il existe des relations entre unités textuelles autres que flexionnelles. Elles sont notamment exploitées en recherche d'information pour augmenter le rappel, soit au cours du requêtage, soit à l'indexation. Ces relations sont exclusivement sémantiques et peuvent en général être typées par les relations sémantiques habituelles comme l'hyponymie/hyperonymie, la méronymie, la synonymie, etc. Pour désigner certaines relations sémantiques entre les unités, il manque une terminologie précise, comme par exemple pour les renvois entre le masculin et le féminin des noms de métier et les liens entre abréviations, sigles et leurs formes développées.

Méthodes

Dans tous les traitements précédents, nous n'avons pas introduit de niveau sémantique. Le lemme n'est pas une unité sémantique, et ne peut donc être utilisé pour représenter des sens. Si on le fait tout de même, il est impossible de faire le lien entre certains lemmes, comme on le voit dans l'exemple suivant : *conductible*, *conduction*, et *conducteur* (figure 17, où les liens sémantiques sont représentés par des flèches). Le lemme est alors utilisé comme représentant de toutes les formes qui lui sont liés dans le lexique.

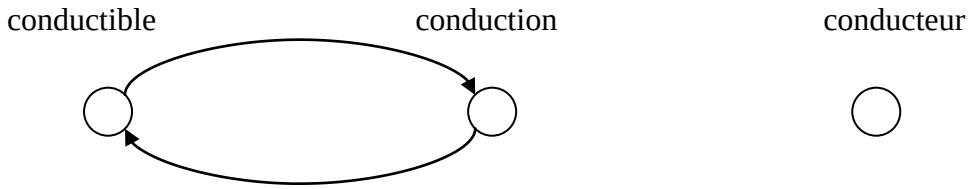


Figure 17 : Graphe de renvois sémantiques entre lemmes

Le lemme *conducteur* est ambigu contrairement aux autres lemmes du graphe : il a un sens en plus qui est celui du conducteur de véhicule. Pour éviter la confusion avec les différents sens, *conducteur* est isolé des autres nœuds.

Pour cet exemple, l'ajout de la catégorie grammaticale résout le problème. En effet, l'ambiguïté du mot conducteur coïncide avec un changement de catégorie. Nous obtenons alors un graphe fortement connexe contenant les 3 éléments (figure 18).

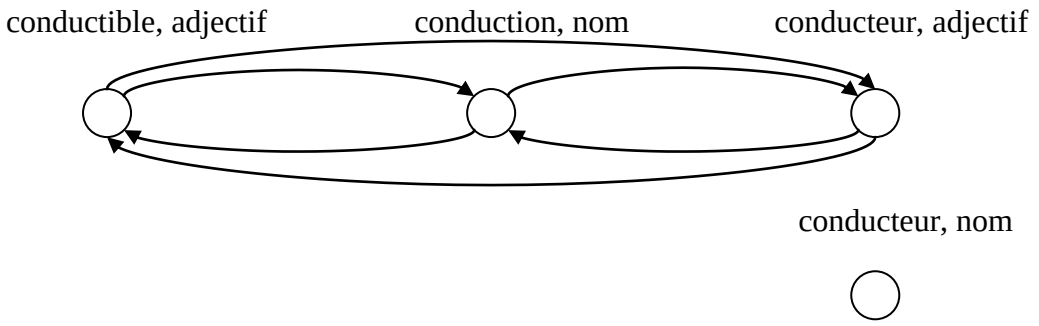


Figure 18 : Graphe de renvois sémantiques entre couples (lemme, catégorie)

L'ajout de la catégorie grammaticale ne suffit néanmoins pas dans de nombreux cas. Ainsi par exemple, *maquisard* est un adjectif dérivé de *maquis*, mais n'en partage qu'un seul des sens, celui qui a trait à la *résistance*. Les renvois doivent donc être faits à partir des sens (figure 19).

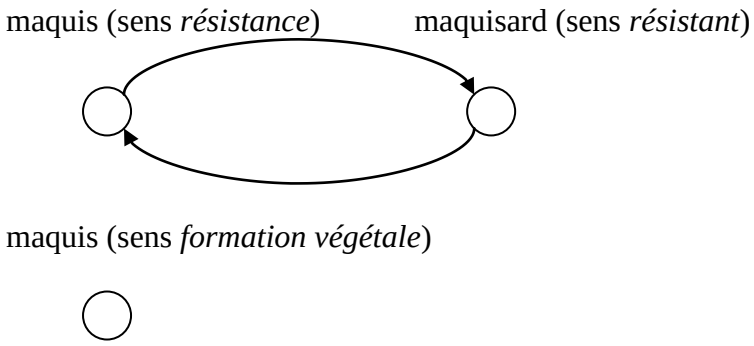


Figure 19 : Graphe de renvois sémantiques entre sens

Un autre exemple est celui de *brasser* et *brassage*. Ces mots ont des sens supplémentaires à *brasserie* et *brasseur*. On voit néanmoins bien qu'il existe un lien morphologique et sémantique entre les différentes formes : ces opérations de construction sont étudiées par la *sémantique dérivationnelle* ou *constructionnelle.* [Dal et al., 2004], qui présente les acquis du projet MorTAL, donnent une bonne introduction en la matière. Les mêmes règles qui servent à l'analyse des unités textuelles (voir 5.4) peuvent être utilisées pour constituer des lexiques de ce type, qui doivent ensuite être corrigés manuellement.

Ce type de données présuppose de pouvoir passer du niveau morphosyntaxique au niveau sémantique, le niveau des sens. Le domaine abordé ici est celui de la désambiguïsation lexicale, mieux connue sous le sigle WSD pour *Word Sense Disambiguation*. [Ide et Véronis, 1998] présentent un excellent état de l'art de l'avancement dans ce domaine jusqu'à la fin du siècle dernier.

Au cœur de cette problématique se trouve le problème dans la distinction entre les sens d'un même mot. Comme on peut lire dans [Kilgarriff, 1997], les tests ne sont pas imperméables et le découpage en sens par des lexicographes est fortement subjectif. Tout découpage est lié à la tâche pour laquelle il est fait : ce qui est un sens dans un type de corpus n'est qu'un simple usage dans un autre. La discussion sur ce qu'est un sens n'est pas tranchée aujourd'hui et pose un problème à la construction de ressources généralistes. Les tentatives d'utilisations des sens tels que codés dans les dictionnaires ont abouti à un échec (voir 4.1.1), et d'autres

propositions de représentation du sens ont vu le jour, comme par exemple les synset de Wordnet.

Pour éviter de tomber dans le piège sémantique des sens lexicaux, on peut dans certains cas s'en tenir à des listes de lemmes. Il faut néanmoins tenir compte des limitations. Cette approche se défend bien si le corpus est spécialisé, car les termes spécialisés y sont moins ambigus. C'est le cas des termes simples, mais encore plus pour les composés. Le même phénomène se produit pour les entités nommées : comme elles sont rarement ambiguës au sein d'un même document voire dans le même corpus, l'absence du passage au niveau sémantique n'est pas bien grave tant qu'on tient compte des limites de l'approche.

L'équivalence est l'un des liens sémantiques courants. Il couvre les synonymes, mais aussi les variantes orthographiques dont voici quelques exemples. Chaque ligne contient des formes équivalentes. En effet, ce lien s'exprime au niveau de la forme et non celle du lemme.

DE Fußbälle, Fußbaelle, Fussbälle, Fussbaelle

Fußböden, Fußboeden, Fussböden, Fussboeden

Fußsprünge, Fußspruenge, Fusssprünge, Fussspruenge

NL kievit, kieviet

kauwgum, kauwgom

PL armagnac, armaniak

FR antimondialisation, anti-mondialisation[81]

L'allemand est connu pour l'agglutination de ses mots composés, mais on sait aussi qu'ils peuvent très bien s'écrire avec un tiret :

DE Der auf dem **Auto-Salon** Genf vorgestellte Crossover X-Bow aus der **Motorrad-Schmiede** KTM schlägt erfolgreich die Brücke zwischen zwei und vier Rädern.

Les synonymes, au contraire, sont idéalement traités au niveau des sens, mais cela pose le problème déjà évoqué. On a l'impression que les variantes sont bien gérées au niveau des mots-formes et des unités textuelles. Les exemples sont peu nombreux, mais il existe des mots dont certains sens ont des variantes et pas les autres. L'exemple suivant est en néerlandais.

NL katoog – kattenoog : variantes pour « œil d'un chat »

katoog : pas de variantes pour le minéral « œil-de-chat »

Cette ambiguïté a aussi des conséquences pour la décomposition des mots composés. Dans le sens d'« œil d'un chat », il conviendrait de décomposer le mot en *kat* (« chat ») et *oog* (« œil »). Pour le sens « œil-de-chat » il ne faudrait surtout pas décomposer le mot.

Un autre exemple en allemand : *die Steuer* (« impôt », nom féminin) et *das Steuer* (« volant », nom neutre) ont beau avoir des paradigmes de flexion différents, dans des noms composés comme *Steuersenkung* (« baisse d'impôts »), on ne peut faire la différence sémantique qu'à travers une décomposition basée sur les sens.

Nous appelons le fait d'appliquer une sémantique sans passer au niveau des sens une *sémantique de surface*.

[81] Aussi bien sur Yahoo.fr que sur Google.fr, les requêtes "antimondialisation" et "anti-mondialisation" donnent presque autant de résultats, respectivement 152 000/151 000 pour Yahoo et 31700/29300 pour Google.

Connaissances

Le traitement fournit une unité u_2 à partir d'une unité u_1 si elles sont liées par le type de relation spécifié.

$$\theta_{rel}(u_1, C_{rel}) = (u_2)$$

Les informations sont liées entre elles par une relation typée *tdr*. Ces informations peuvent être des unités textuelles (qui n'existent pas forcément comme mot-forme dans les connaissances), des mots-formes ou des lemmes avec ou sans leur description.

$$C_{rel} = \{rel_1, rel_2, ..., rel_n\}$$

où $rel = (uRel, tdr, uRel)$,

où $uRel = (f_1, d_1)$ ou $uRel = (lem, d, cf)$

5.8 Etiquetage sémantique

Objectif

L'étiquetage sémantique consiste à rattacher des informations d'ordre sémantique aux unités textuelles afin de les exploiter dans les calculs de pertinence ou dans les grammaires de détection des entités.

Méthodes

On peut enrichir le texte avec des informations sémantiques à plusieurs niveaux du texte. Le plus évident est de rattacher des informations directement aux unités textuelles. On peut par exemple indiquer qu'une certaine unité est une ville, une couleur, un jour de la semaine, un nom d'entreprise ou un prénom. On peut aussi lui attribuer une ou plusieurs thématiques, comme par exemple *météo, bourse* ou *métallurgie*. Dans tous les cas, il s'agit d'informations prédéfinies qui sont obligatoirement puisées dans une source de connaissances. L'étiquetage sémantique se distingue ainsi des traitements précédents, car le traitement ne peut se faire uniquement à partir de règles qui s'appliquent sur les unités textuelles.

Comme pour la mise en relation entre unités (décrite dans le paragraphe précédent), il serait préférable de disposer d'unités sémantiques plutôt que d'attribuer des descriptions sémantiques à des unités textuelles (en direct ou par l'intermédiaire d'une description calculée comme c'est notre cas).

Cela nécessiterait un passage au niveau sémantique tel que pratiqué en WSD. En recherche d'information, l'utilité de cette approche n'a pas été prouvée en pratique, même si intellectuellement elle semble prometteuse. C'est pour cette raison que deux tâches spécifiquement ciblée sur le possible apport de la désambiguïsation lexicale ont été organisées à CLEF 2008 : *Robust WSD Task*[82] et *QA-WSD*[83]. Pour la tâche de recherche d'information, les organisateurs estiment en effet que la polysémie est à l'origine de l'échec des systèmes de recherche d'information et veulent mesurer l'apport de l'introduction des techniques de

[82] Word Sense Disambiguation for (cross-lingual) Information Retrieval : http://ixa2.si.ehu.es/clirwsd/
[83] Word Sense Disambiguation for Question Answering : http://ixa2.si.ehu.es/qawsd/

désambiguïsation lexicale[84]. Les données sémantiques exploitées sont celles de WordNet. Les résultats de la *Robust WSD Task* sont mitigés, certains annonçant une amélioration pour certains types de requêtes ([Martínez-Santiago et al., 2008] et [Navarro et al., 2008]), d'autres pour la recherche multilingue mais pas monolingue comme [Otegi et al., 2008]. [Guyot et al., 2008] estime même que, si WSD en RI peut marcher dans certains cas très précis comme les requêtes courtes, le traitement ne sert pas à grand-chose dans la plupart des cas : la complexité de la compréhension de la requête dépasse largement le champ du WSD. En question réponse, [Buscaldi et Rosso, 2008] n'ont constaté aucun apport avec inclusion de WSD, mais les auteurs affirment que le corpus livré ne se prêtait pas à la tâche de question-réponse : 75 % des questions n'auraient pas eu la réponse dans le corpus. Le succès de la participation à la *Robust WSD Task* a néanmoins mené à l'organisation d'une nouvelle édition à CLEF 2009. Les résultats étaient similaires à ceux de l'année précédente : certains systèmes de recherche d'information constatent une légère amélioration, mais les meilleurs résultats sont obtenus par les systèmes qui n'utilisent pas de WSD ([Agirre et al., 2010]).

L'apport de la WSD en RI ne fait donc pas (encore) l'unanimité. La mise en œuvre d'une sémantique plus superficielle, qui ne s'applique pas sur des sens mais sur les unités textuelles ou à partir de leurs analyses, est une approche courante. Des heuristiques peuvent être utilisées pour essayer de désambiguïser certains types d'unités. La capitalisation peut par exemple être utilisée pour distinguer les prénoms de leurs homographes non capitalisés, sauf en début de phrase bien sûr.

En absence du sens comme élément raccrocheur, il faut bien en choisir un autre: cela peut être la forme, le lemme, le lemme avec la catégorie grammaticale, ou encore plus précis: lemme, catégorie grammaticale et classe de flexion. Cela dépend d'une part des traitements précédents tels que prévus dans l'architecture du système, et d'autre part de la précision qu'on veut obtenir. Comme nous l'avons vu auparavant (5.6) avec l'exemple du mot portugais *ponteiro* qui est à la fois nom et adjectif, ou du mot allemand *Steuer*, qui est à la fois masculin et féminin mais avec des sens différents, plusieurs degrés de précision sont possibles et parfois nécessaires. Nous avons pu constater la viabilité de cette approche dans la partie *recherche multilingue* du projet Vodel, où nous avons expérimenté une sémantique superficielle accrochée aux lemmes[85].

Au niveau de la ressource, le lexique sémantique ne peut être mélangé directement avec les lexiques morphosyntaxiques, à moins d'appliquer d'autres critères de conception lexicale pour les informations sémantiques que pour les informations morphosyntaxiques. L'ajout d'une ambiguïté morphosyntaxique se fait par un ajout d'une entrée <mot-forme, description> dans le lexique. L'utilisation du même critère pour l'ajout d'entrées sémantiques mènerait à une prolifération des entrées du lexique, ajoutant un nombre d'entrées linéaire égale au produit du nombre de formes du lemme concerné et le nombre d'ambiguïtés sémantiques ajoutés. Il faut donc séparer les deux lexiques pour préserver la cohérence du lexique morphosyntaxique, à moins de complexifier la structure du lexique à tel point qu'il faut un outil d'édition. Avec plusieurs fichiers, le désavantage est qu'on assiste à une duplication de certaines informations sur plusieurs sources, au grand dam des personnes qui gèrent les lexiques.

[84] "The goal of the task is to test whether WSD can be used beneficially for retrieval systems. The organizers believe that polysemy is among the reasons for information retrieval (IR) systems to fail. WSD could allow a more targeted retrieval." (http://ixa2.si.ehu.es/clirwsd/)

[85] Voir [Cailliau et al., 2008] pour une description complète.

Connaissances

Le traitement décrit ici donne au moins un trait sémantique *tS* à une unité sémantique *uS* que nous avons choisie de représenter comme un triplet de lemme (*lem*) , de catégorie grammaticale (*cat*) et de classe de flexion (*cf*), résultat des traitements décrits précédemment.

$$\theta_{sem}(uS, C_{sem}) = \{dS_1, dS_2, ..., dS_n\}$$

où $dS = (tS_1, tS_2, ..., tS_m)$

$uS = (lem, cat, cf)$

et $n \geq 1$

Les connaissances sémantiques sont un ensemble de couples d'unités sémantiques *uS* et de descriptions sémantiques *dS*. L'unité sémantique et la description sémantique correspondent aux équivalents choisis dans le traitement.

$$C_{sem} = \{(uS_1, dS_1), (uS_2, dS_2), ..., (uS_n, dS_n)\}$$

où $uS = (lem, cat, cf)$

$dS = (tS_1, tS_2, ..., tS_n)$

5.9 Détection des entités

Objectif

La détection des entités consiste à détecter dans un flux d'information des entités prédéfinies, notamment les entités nommées comme les noms de personnes, noms géographiques, noms d'organismes, les mesures et prix, les adresses, etc., mais aussi des groupes nominaux courts.

Méthodes

Les informations exploitées pour la détection des entités peuvent être très différentes selon la méthode adoptée. Elles portent typiquement sur des propriétés de la forme (par exemple la capitalisation), morphologiques (par exemple le nombre) et sémantiques (les traits de l'étiquetage sémantique). Les méthodes par apprentissage sont connues pour être très dépendantes du corpus d'apprentissage, comme nous l'avons déjà pointé plus haut dans le cadre de la désambiguïsation morphosyntaxique (5.5). Les systèmes à base de règles linguistiques sont réputés plus robustes quand ils sont confrontés à des textes dont le type ne faisait pas partie du corpus d'étude, mais [Poibeau et Kosseim, 2001] signalent tout de même une chute de 90 % à 50 % quand ils sont confrontés à un genre de texte plus informel. On ne sait pas en même temps quelle aurait été la chute d'un modèle à base d'apprentissage. En introduisant d'autres stratégies, comme l'augmentation de la taille de dictionnaire et un système d'auto-apprentissage, les mêmes auteurs arrivent à remonter le score jusqu'à 84 %.

Le sujet des *shared task* des éditions 2002 et 2003 de CoNLL, la *Conference on Computational Natural Language Learning*, était justement la détection des entités nommées avec des méthodes « indépendantes des langues ». La tâche en 2002 ([Tjong Kim Sang, 2002]) prenait comme objet l'espagnol et le néerlandais et fournissait des données d'apprentissage contenant seulement les tokens et les tags d'entités nommées. Les entités à détecter étaient de quatre types : personnes, endroits, organisations et « autres entités ne faisant pas partie des trois précédents ». Le meilleur système obtenait une précision et un rappel d'un peu plus de 81 % en espagnol, alors que les scores sont moins bons de quelques

points en néerlandais, respectivement de 78 % et de 76 %. En 2003 ([Tjong Kim Sang et De Meulder, 2003]), les langues étaient l'anglais et l'allemand et des données non annotées ont été fournies. En plus, d'autres sources d'information comme des index de noms propres (des *gazetteer*) pouvaient être utilisés. Le meilleur système en anglais obtient près de 89% en précision et en rappel. Le même système est également le meilleur en allemand, avec près de 84% en précision et 64% en rappel. Ces chiffres sont les moyennes pour les quatre types d'entités. Le meilleur système, décrit dans [Florian et al., 2003], prend en compte les résultats de plusieurs classifieurs, de deux autres systèmes de détection d'entités ainsi que les informations en provenance de larges index de noms propres anglais. En examinant les résultats détaillés par langue et par entité, on remarque que la précision de la détection des noms de personne est à peu près pareille dans les deux langues (autour de 92 %, mais que le rappel chute de 20 points en allemand passant de 95 % à 75 %. La chute en rappel est au moins aussi violente pour les autres entités. Les raisons données sont la capitalisation des noms communs et la légère couverture du gazetteer par rapport à l'anglais.

Depuis le temps que la détection d'entités nommées fait partie des sujets traités en TAL, on pourrait croire que c'est un sujet clos. En 1997 le score du meilleur système dans MUC-7 était de 93,39 tandis que le « plus mauvais » annotateur humain obtenait 96,95 ([MUC7, 1997]). L'organisation des campagnes d'évaluation ACE (*Automatic Content Extraction*) par le NIST en 2007 et 2008 ([ACE, 2008]) montre que les entités nommées restent un sujet de la plus grande importance. Aujourd'hui la robustesse des systèmes devient un facteur d'importance. Il ne s'agit plus de se doter de ressources les langues économiquement moins importantes comme c'est le cas avec l'organisation du second concours HAREM en 2008 pour l'extraction des entités nommées en portugais ([Mota et Santos, 2008]). Les corpus d'évaluation d'ACE 2007 EDR (*Entity Detection and Recognition*) pour la langue anglaise étaient formés de sources écrites ainsi que de transcriptions de différentes sources. Voici les résultats des deux meilleurs systèmes[86]. La liste des entités à reconnaître était la suivante : personnes, organisations, endroits, facilités, entités géopolitiques, armes, véhicules.

Site	Overall	Broadcast Conversa-tions	Broadcast News	Newswire	Telephone	Usenet	Web-logs
BBN Technologies	56.3	44.7	65.4	58.1	49.2	39.2	52.7
IBM	52.7	48.7	65.9	52.8	45.4	44.0	45.8

Tableau 7: Deux meilleurs résultats de ACE 2007

D'après ces chiffres, on peut dire qu'il reste globalement une bonne marge de progrès. On voit également qu'il existe des fortes disparités entre les différents types de texte, et que les meilleurs scores sont obtenus pour les journaux radiophoniques, ce qui était d'ailleurs le cas pour tous les autres systèmes.

Les entités cherchées dans ACE 2008 étaient : personnes, organisations, endroits, facilités, et entités géopolitiques. Les genres de texte y sont encore un peu plus divers.

[86] Consultable sur : http://www.nist.gov/speech/tests/ace/2007/doc/ace07_eval_official_results_20070402.html

Site	Overall	Broadcast Conversations	Broadcast News	Meetings	Newswire	Telephone	Usenet	Weblogs
# IBM	50.8	44.6	37.7	-11.9	58.1	26.1	25.5	51.0
BBN Technologies	52.6	42.0	36.9	-44.2	61.3	22.1	31.1	54.8

Tableau 8 : Deux meilleurs résultats de ACE 2008 (LEDR)

Ces scores montrent combien il est difficile d'obtenir un score correct en détection d'entités nommées. A titre d'exemple *Gap* est une marque, une entreprise, et une commune française de taille moyenne. *Orange* est un fruit, une couleur, une marque, une entreprise, et une commune française de taille moyenne. L'exemple suivant est un titre d'article issu d'internet[87]. Y apparaît une famille qui s'appelle *Boring*, ce qui en anglais veut dire ennuyeux.

> Boring couple forces Google into admitting that complete privacy no longer exists
>
> … The crime in question being the photographing by Google of the Boring house. …

Si cet exemple illustre toute la difficulté de la désambiguïsation des noms de famille, cela reste du texte bien écrit dans un format (html) dont le texte est relativement facile à extraire. Ce n'est pas toujours le cas quand les documents d'origine ont des formats propriétaires, ce qui est généralement le cas pour des documents d'entreprise. Dans ce cas l'extraction du texte peut être moins bonne. Les colonnes dans un document PDF ne sont pas forcément reconnues comme telles. Le résultat est que l'on se retrouve avec des phrases syntaxiquement incorrectes et des mots inconnus à cause des césures.

Ces problèmes d'extraction ont de lourdes conséquences sur les règles de détection d'entités. Il est bien plus difficile de garantir la robustesse sur ces documents. Des règles plus robustes qui privilégient la précision dans toutes les circonstances font inévitablement chuter le rappel. La solution idéale serait d'avoir des jeux de règles selon la grammaticalité du texte extrait, mais cette approche est très coûteuse pour l'analyse des documents ainsi que pour l'écriture des règles. C'est d'ailleurs la solution que préconisent les auteurs de [Maynard et al., 2001] sans qu'ils sachent la mettre en œuvre automatiquement.

Notre expérience s'articule principalement autour de quatre types d'entités : les groupes nominaux courts (avec des restrictions sémantiques), les noms de personnes, les noms géographiques et les noms d'entreprises. Des stratégies différentes correspondent à chacune de ces entités.

Le formalisme que nous connaissons le mieux est celui des automates à états finis. C'est également celui adopté par Intex/Unitex ([Silberztein, 1993]) et Annie, le module de détection d'entités de GATE pour encoder des règles linguistiques. Un bon récapitulatif des bases formelles des automates à états finis en TAL peut être trouvé dans [Mohri, 1997]. Cette approche exploite différentes propriétés de la forme (éventuellement sous forme d'expression régulière) et de la description telles que le lemme, la catégorie grammaticale et les traits morphologiques. Elle exploite également des informations sémantiques de surface provenant d'un étiquetage sémantique superficiel. Cet étiquetage sémantique doit être vu comme une hypothèse : ce n'est qu'après la détection d'un nom de personne qu'une unité textuelle peut être identifiée comme étant un prénom. Comme décrit précédemment, cette sémantique de

[87] Copie d'écran de la page internet en Annexe C.

surface nécessite des sources lexicales. Selon le type d'entité considéré le nombre d'informations requises peut être plus ou moins important.

Ces ressources sont rarement parfaites. Vu la nature et l'étendue de ce qu'on essaie de traiter, les informations ne peuvent être complètes. Parfois les informations sont trop étendues, comme des prénoms peu habituels dans un lexique de prénoms. A cause de ce double problème, générant respectivement silence et bruit, les règles appliquées peuvent se complexifier considérablement. Ce problème n'est soluble que si le corpus est connu d'avance, ce qui est rarement le cas. Un système robuste doit donner de bons résultats dans toutes les situations.

Les entités ont été très bien définies dans le cadre de ACE, mais ces définitions ne sont pas toujours celles des applications industrielles. Pour celles-ci la précision doit être la plus haute possible, ce qui mène à la reconnaissance d'un sous-ensemble des entités telles qu'elles sont définies dans ACE. Ceci est motivé par l'état d'esprit des clients, à qui il est plus facile d'expliquer un cas de silence qu'un cas de bruit (sous réserve que les règles soient logiques évidemment).

Comme dans d'autres traitements, c'est l'ambiguïté, caractéristique de la langue, qui pose le plus de problèmes. Les prénoms, les noms de familles, les noms géographiques, les noms d'entreprises, etc. peuvent tous être ambigus. Cette ambiguïté existe parfois entre des mots qui ne sont pas de mots communs, comme entre un prénom et un nom de famille (ex. *Reagan, Martin*) ou encore une marque et un nom de famille (*Renault*). En comparant des lexiques de prénoms ou de marque avec des lexiques de large couverture on peut déceler une bonne partie des ambiguïtés, mais cela reste toujours imparfait.

La normalisation sert à mettre en correspondance plusieurs entités dont la forme de surface est différente, comme par exemple les entités suivantes :

FR Guillem Louis Marie de Blancfort
Guillem L. M. de Blancfort
Guillem de Blancfort
de Blancfort Guillem
G. de Blancfort
M. de Blancfort
Président de Blancfort

FR Kwaga SA
Kwaga

ES México
Méjico

ES La Habana
San Cristóbal de La Habana
Ciudad de La Habana.

La forme normalisée est une des formes de surface et c'est celle qui est affichée dans l'interface.

Parfois les cas sont difficiles à trancher, comme par exemple la différence entre *IBM* et *IBM France*. Nous le considérons comme des entités différentes, mais selon les applications on peut juger différemment. Le repérage de la coréférence des entités qui désignent les mêmes entités peut se faire par des règles, mais demande parfois également des ressources lexicales. Pour les personnes par exemple, on peut en général se contenter de règles, tandis que pour les noms géographiques ou les entreprises il faut des ressources en plus.

Le fait de normaliser ces entités fait tendre ce traitement d'une analyse sémantique de surface vers une sémantique profonde. En effet, ces entités deviennent de vraies entités sémantiques grâce à la normalisation. Mises en relation elles peuvent servir à créer un graphe de réseau social ou de compétences. Le piège de la WSD n'est pas tendu car nous traitons ici des noms propres ou des groupes nominaux qui ne sont pas ambigus. La forme normalisée représente donc une vraie information sémantique à exploiter.

Connaissances

La détection des entités exploite les descriptions étendues *Dent* des unités *u* fournies par l'ensemble des traitements précédents, notamment les descriptions morphosyntaxiques *Dm* et sémantiques *Ds*. En sortie on obtient des suites d'unités textuelles annotées avec le type d'entité détectée. Si une lemmatisation a été effectuée depuis l'analyse et l'étiquetage morphosyntaxique, la description morphosyntaxique *dM* a été enrichie d'un lemme, qui est une connaissance utile pour la détection des entités, d'où l'ajout de *lem* dans *dM*.

$$\theta_{ent}(((u_1, Dent_1),(u_2, Dent_2),...,(u_n, Dent_n)), C_{ent}) =$$
$$\{((u_i,...,u_{i+m})_1, type),((u_j,...,u_{j+k})_2, type),...,((u_l,...,u_{l+o})_h, type)\}$$

où $Dent = \{dEnt_1, dEnt_2,..., dEnt_n\}$,

$i, j, l \geq 1$,

$m, k, o \geq 0$,

$dEnt = (dM, dS)$,

$dM = (f, lem, Tm)$

et $C_{ent} = Gent = \{rEnt_1, rEnt_2,..., rEnt_n\}$

5.10 Conclusion

Si les traitements que nous avons modélisés dans ce chapitre sont tous déployés dans notre système, ils ne sont pas pour autant spécifiques à la recherche d'information Certains sont de bas niveau et reviennent dans chaque application qui est confrontée à un traitement de la langue. Les autres peuvent également être appliqués dans d'autres contextes que la recherche d'information. Il est impossible d'être exhaustif dans les traitements : nous avons pris comme critère de sélection l'existence dans le système que nous avons étudié. Néanmoins, tout traitement peut se modéliser de la même façon.

Nous avons fait le point sur les différentes méthodes pour mettre en œuvre ces traitements. L'approche la plus précise est souvent celle qui utilise intensivement des ressources. En revanche, cette meilleure précision s'accompagne d'un surcoût lié à la gestion des ressources exploitées. En raison de la quantité de connaissances à gérer, ce coût peut être important, notamment dans des systèmes multilingues : les ressources sont spécifiques à la langue et donc à développer pour chaque langue, alors qu'il peut exister d'autres approches plus *génériques*, moins coûteuses, mais peut-être moins précises ou moins facilement corrigibles.

Maintenant que nous savons quelles connaissances sont requises par chaque traitement de notre système, nous avons créé le cadre nécessaire pour modéliser l'architecture linguistique de notre système et ainsi faire le lien avec les informations que contiennent nos ressources.

Chapitre 6

Modéliser l'architecture linguistique d'un système

En nous appuyant sur le cadre créé dans le chapitre précédent, nous modélisons dans ce chapitre l'architecture linguistique de l'analyseur linguistique du moteur de recherche de Sinequa. L'architecture linguistique décrit les objets et les relations entre les objets du système. Nous décrivons donc d'abord les traitements en indiquant les connaissances prises en compte et les informations produites. Ensuite nous décrivons et formalisons les ressources sur lesquelles reposent ces traitements. Avant de formaliser chaque ressource, nous fournissons une brève fiche descriptive et nous faisons suivre la formalisation par un rendu visuel en UML. Toutes les ressources directes ou indirectes du système sont formalisées. Les corpus de validation ne font pas partie des ressources, la validation n'étant pas un traitement du système. L'architecture linguistique est indépendante des langues : elle s'applique sur l'ensemble des 19 langues de notre système qui sont représentatives de la plupart des familles de langues du monde. Les traitements sont les mêmes pour toutes, seules les ressources linguistiques diffèrent.

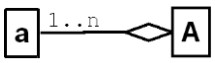
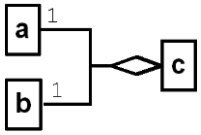
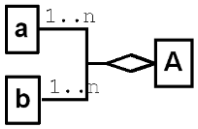
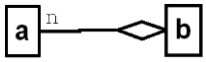
6.1 Rendu visuel de l'architecture linguistique

Nous décrivons les traitements et leurs ressources en utilisant la théorie des ensembles. Dans des travaux précédents publiés dans [Cailliau, 2006], nous avions choisi UML comme langage de représentation des connaissances. Or, UML n'étant pas motivé par un langage logique, on ne peut pas l'utiliser pour la construction d'un modèle. Il ne nous servira donc que pour illustrer notre modèle. UML est également utilisé dans les travaux de spécification de la norme ISO (voir 4.2.5, p. 79)

Pour la représentation graphique du modèle, nous avons choisi un sous-ensemble du langage UML ([Booch et al., 1998]). Les avantages de ce langage sont qu'il est intuitif, visuel, facile à appréhender et adapté à la modélisation de la langue. En outre, il est utilisé dans la norme ISO 24613:2008 (voir 4.2.5) pour la modélisation lexicale. L'inconvénient est qu'UML est un langage de modélisation graphique qui n'est pas motivé mathématiquement[88]. Il est donc impossible de raisonner à partir d'un modèle UML. Nous tirons seulement profit des capacités de rendu visuel pour visualiser un modèle utilisant la théorie des ensembles.

Le sous-ensemble du langage utilisé se limite aux diagrammes de classes avec seulement trois types de relations entre les classes : la généralisation, l'agrégation et une association nommée *se* pour *sous-ensemble de*. L'agrégation comporte une cardinalité qui indique le nombre d'instances que peut comporter la classe agrégante.

Le passage de la représentation ensembliste vers UML est fait selon les correspondances indiquées dans le tableau 9. En plus, chaque élément d'ensemble et chaque ensemble comme des classes. Celles-ci sont représentées par des rectangles qui portent le nom de la classe.

	Ensembles	Relation UML	Représentation UML
1	$A = \{a_1, a_2, ..., a_n\}$	Agrégation	
2	$c = (a,b)$	Agrégation	
3	$A = \{(a,b)_1, (a,b)_2, ..., (a,b)_n\}$	Agrégation	
4	$b = (a_1, a_2, ..., a_n)$	Agrégation	

[88] Depuis la spécification d'UML, il existe des recherches sur la façon de formaliser les structures sous-jacentes de langages de modélisation graphiques. Certains de ces travaux cherchent à établir une notation graphique qui est formalisable comme [Fradet et al., 1999] pour pouvoir ensuite raisonner dessus, alors que d'autres, comme [France et al., 1998] et [Evans, 1998] cherchent à formaliser UML même. Les auteurs de ces deux derniers articles ont même initié le groupe *pUML, the precise UML group*, pour développer un modèle sémantique précis pour les diagrammes UML (voir http://www.cs.york.ac.uk/puml). Nos travaux ne cherchent pas à donner une légitimité sémantique à UML.

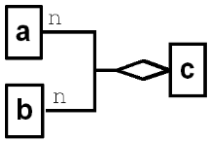
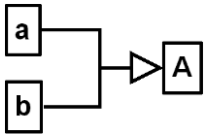
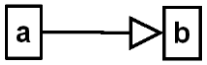
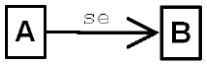
5	$c = ((a,b)_1,(a,b)_2,...,(a,b)_n)$	Agrégation	
6	$A = \{a,b\}$	Généralisation	
7	$a \in A, b \in B, A \subset B$	Généralisation	
8		Sous-ensemble de	

Tableau 9 : Correspondances représentations Ensembles – UML

Ce tableau de correspondance est indispensable pour l'interprétation des diagrammes UML, sinon on pourrait par exemple interpréter la représentation UML du 3 dans le tableau 9 comme un ensemble de deux ensembles de n éléments : $A = \{\{a_1,a_2,...,a_n\},\{b_1,b_2,...,b_n\}\}$, et non comme un ensemble de *n* listes.

Il aurait été intéressant de faire la distinction entre les relations d'agrégation et de composition. La composition peut être vue comme une agrégation forte : les opérations sur l'agrégat d'une composition affectent aussi les composants, ce qui n'est pas le cas de l'agrégation. Malheureusement, notre modèle ensembliste ne fait pas cette distinction. Les classes d'une composition ne peuvent pas faire partie en même temps de plus d'un agrégat, ce qui n'est pas notre cas non plus.

UML est un langage de modélisation graphique orienté objet. Les propriétés communes des objets mis en relation sont de préférence abstraites dans des classes plus générales, simplifiant ainsi la description des objets héritants. Dans certains cas, ce mécanisme nous a obligé à introduire de nouveaux objets qui n'existaient pas explicitement dans les ensembles. Il aurait été possible de réadapter le modèle ensembliste, mais cela aurait affaibli le modèle original. A chaque fois qu'il y a ce genre d'ajouts, nous l'indiquons par *Ajouts de composants structurels pour la visualisation*.

La relation entre notre modèle ensembliste et sa visualisation graphique n'est donc pas aussi évidente que nous l'aurions souhaitée. L'interprétation que nous donnons aux diagrammes UML n'est pas la seule possible, et les diagrammes auraient sans aucun doute pu être dessinés différemment tout en correspondant au même modèle exprimé. Nous aurions pu inventer un nouveau langage en nous inspirant des réseaux d'héritage ([Daelemans et al., 1992]) et en ajoutant d'autres relations. Nous préférons néanmoins rester dans un cadre existant, bien connu et répandu, même si cela entraîne les inconvénients mentionnés.

6.2 Architecture des traitements

Les traitements reposent sur les ressources ainsi que sur les informations produites par les traitements qui les précèdent. Nous appelons *C* les informations contenues dans les ressources, et *K* celles qui sont produites par les traitements. Dans ce qui suit, nous décrivons

brièvement les traitements du système et les *K* qu'ils produisent. La « dépendance K » indique que le traitement en cours se base sur les *K* produites par le traitement indiqué. Les connaissances *C* sont indiquées par leur symbole et sont formalisées en 6.3.

- **Génération de lexique** θ_{gen}

Génère des mots-fomes avec leur description morphosyntaxique.

Dépendance K : aucune.

K : lexique général

C : Lg, Gflex, Ggen

- **Identification de la langue** θ_{idl}

Découpe le flux en fragments.

Dépendance K : aucune.

K : fragments monolingues et langue du fragment

C : Lmfr, Lngram

- **Découpage en unités textuelles** θ_{dec}

Découpe le fragment monolingue en unités textuelles.

Dépendance K : θ_{idl}, θ_{gen}

K : fragment découpé en unités textuelles

C : Lg

- **Phonétisation** θ_{phon}

Donne une ou plusieurs formes phonétiques à une unité textuelle.

Dépendance K : θ_{dec}

K : forme phonétique pour chaque unité textuelle

C : Gphon

- **Analyse et étiquetage morphosyntaxique des unités textuelles** θ_{etiq}

Analyse chaque unité textuelle et donne une description morphosyntaxique.

Dépendance K : θ_{dec}

K : une ou plusieurs descriptions morphosyntaxiques pour chaque unité textuelle, que celle-ci soit dans le lexique ou non

C : Lg, Gm, Gcomp

- **Désambiguïsation morphosyntaxique** θ_{des}

Diminue le nombre d'ambiguïtés morphosyntaxiques d'une unité textuelle en fonction de son contexte.

Dépendance K : θ_{etiq}

K : une ou plusieurs descriptions morphosyntaxiques pour chaque unité textuelle

C : ML (= GdesT, Ldes), GdesR

- **Lemmatisation** θ_{lemm}

Fournit un lemme pour chaque unité textuelle

Dépendance K : θ_{des}

K : lemme pour chaque analyse morphosyntaxique

C : Lg

- **Renvois entre unités textuelles** θ_{rel}

Met en relation deux unités textuelles

Dépendance K : θ_{lemm}

K : une unité textuelle différente pour les unités textuelles référencées

C : Lvar, Labrev, Lnom, Lrrecip, Lrsimple

- **Etiquetage sémantique** θ_{sem}

Fournit une analyse thématique pour chaque lemme

Dépendance K : θ_{lemm}

K : une ou plusieurs descriptions thématiques pour les lemmes référencés

C : Ls

- **Détection d'entités et de relations** θ_{ent}

Repère des entités à partir de motifs exprimés.

Dépendance K : θ_{lemm}

K : le type d'entité d'une ou d'une suite d'unités textuelles ou le type de relation entre une entité et une ou une suite d'unités textuelles

C : Gent, Gnorm, Gincl

6.3 Modélisation des ressources du système

Les types de ressources du système sont les suivantes, toutes langues confondues :

$$C = \{L, Cor, G\}$$

où $L = Lg \cup Ls \cup Lvar \cup Labrev \cup Lnom \cup Lrrecip \cup Lrsimple \cup Lnorm \cup Lngram \cup Lmfr \cup Ldes$

$$Cor = corML \cup corNgram$$

$$G = Gint \cup Ggen \cup Gm \cup Gcomp \cup Gphon \cup Gflex \cup GdesT \cup GdesR \cup Gincl$$

Les corpus de validation (corVal) ne sont pas compris dans les corpus, car comme indiqués, ce sont des ressources de validation et non pas du système.

6.3.1 Lexique

Nous avons recensé deux types de ressources lexicales : les *lexiques d'analyse* qui sont utilisés pour l'analyse du texte lors de l'indexation et l'analyse de la requête, et les *lexiques d'expansion* de la requête. Il existe un autre type de fichier qui n'est pas une ressource mais un fichier de gestion. Nous l'indiquons comme *metadonnées*.

- **Lexique morphosyntaxique général**

Clé d'entrée	Mot-forme
Informations par clé	Une ou plusieurs descriptions morphosyntaxiques, c'est-à-dire une ou plusieurs combinaisons de : catégorie grammaticale, éventuellement des traits morphosyntaxiques, un lemme et éventuellement des traits sémantiques de surface. Dans les langues compositionnelles, la décomposition de certains mots composés peut être indiquée par des traits et/ou par des lemmes.
Format	Texte formaté.
Type	Lexique d'analyse
Remarques	Peut être éclaté sur plusieurs fichiers pour faciliter la gestion, par exemples les prénoms ou les villes de France. Peut contenir des éléments non autonomes pour les opérations de dérivation, décomposition ou agglutination.
Exemple	FR : confrérie des chevaliers du Tastevin : confrérie_des_chevaliers_du_Tastevin, nom, féminin, singulier, nom propre, mot composé

Formalisation :

$$Lg = \{uL_1, uL_2, ..., uL_n\} \cup \{uAgg_1, uAgg_2, ..., uAgg_n\} \cup \{uDer_1, uDer_2, ..., uDer_n\}$$
$$\cup \{uComp_1, uComp_2, ..., uComp_n\}$$

où $uL = (f, lem, cat, Dm, Ds, Dint, dComp, Dcomp)$,

$uAgg = (e, Dagg)$

$uDer = (e, Dder)$

$uComp = (e, Dcomp)$

où $cat \in Cat$

$Dm \subset Tm$,

$Ds \subset Ts$

$dComp = (lem_1, lem_2, ..., lem_n)$

$Dint \subset Tint$

$Dcomp \subset Tcomp$

$Dagg \subset Tagg$

$Dder \subset Tder$

$Cat = \{cat_1, cat_2, ..., cat_n\}$

$Tm = \{tM_1, tM_2, ..., tM_n\}$

$Ts = \{tS_1, tS_2, ..., tS_n\}$

$Tder = \{tDer_1, tDer_2, ..., tDer_n\}$

$Tagg = \{tAgg_1, tAgg_2, ..., tAgg_n\}$

$Tcomp = \{tComp_1, tComp_2, ..., tComp_n\}$

$Tint = \{tInt_1, tInt_2, ..., tInt_n\}$

Objets :

cat	catégorie grammaticale
Cat	ensemble des catégories grammaticales déclarés
dcomp	description compositionnelle en lemmes
Dagg	description agglutinationnelle (traits)
Dcomp	description compositionnelle en traits
Dder	description dérivationnelle (traits)
De	description morphologique
Dm	description morphosyntaxique
Ds	description sémantique
e	élément morphologique (non-autonome)
f	mot-forme (autonome)
lem	lemme
Lg	Lexique morphosyntaxique général
tagg	trait d'agglutination
Tagg	ensemble des traits d'agglutination déclarés

tcomp	trait de composition
Tcomp	ensemble des traits de composition déclarés
tder	trait de dérivation
Tder	ensemble des traits de dérivation déclarés
tInt	trait d'interprétation
Tint	ensemble des traits d'interprétation déclarés
tM	trait morphosyntaxique
Tm	ensemble des traits morphosyntaxiques déclarés
tS	trait sémantique
Ts	ensemble des traits sémantiques déclarés
uAgg	unité d'agglutination
uComp	unité de composition
uDer	unité de dérivation
uL	unité lexicale

Rendu visuel :

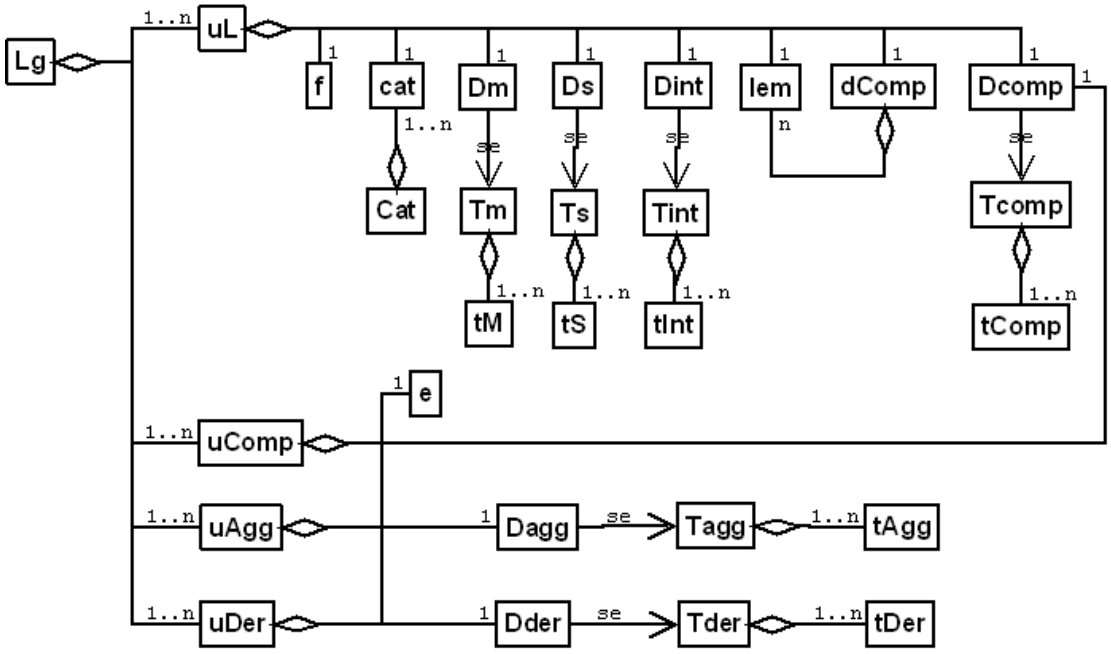


Figure 20 : Diagramme du *Lexique général*

- Lexiques morphosyntaxiques spécifiques

Clé d'entrée	Idem *Lexique morphosyntaxique général*, mais avec des entrées spécifiques liées à un client ou à un domaine de spécialité par fichier.
Informations par clé	
Format	
Type	
Remarques	
Exemple	NL : Verboden Vrucht : Verboden_Vrucht, nom, nom propre, singulier, marque déposée

Formalisation et objets : idem *Lexique morphosyntaxique général*.

- **Référentiel d'étiquettes**

Clé d'entrée	Catégorie grammaticale, trait morphosyntaxique ou trait sémantique
Informations par clé	La présence de la clé est autosuffisante. Elle est accompagnée d'une définition avec un exemple.
Format	Texte formaté.
Type	Métadonnées
Exemple	NOM : nom, … ; MDEP : marque déposée, …

Formalisation :

$$RefM = \left\{ \begin{array}{l} \{(cat_1, def_1), (cat_2, def_2), ..., (cat_n, def_n)\}, \\ \{(tM_1, def_1), (tM_2, def_2), ..., (tM_n, def_n)\}, \\ \{(tS_1, def_1), (tS_2, def_2), ..., (tS_n, def_n)\}, \\ \{(tDer_1, def_1), (tDer_2, def_2), ..., (tDer_n, def_n)\}, \\ \{(tAgg_1, def_1), (tAgg_2, def_2), ..., (tAgg_n, def_n)\}, \\ \{(tComp_1, def_1), (tComp_2, def_2), ..., (tComp_n, def_n)\}, \\ \{(tInt_1, def_1), (tInt_2, def_2), ..., (tInt_n, def_n)\} \end{array} \right\}$$

Objets :

def	définition
RefM	référentiel d'étiquettes

Rendu visuel :

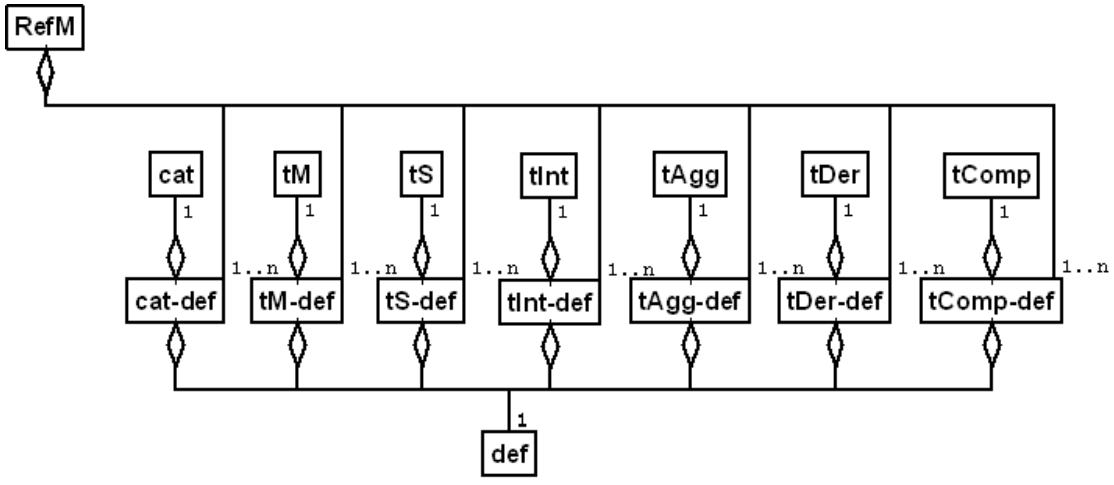


Figure 21 : Diagramme du *Référentiel morphosyntaxique*

Ajouts de composants structurels pour la visualisation :

cat-def, tM-def, tS-def, tInt-def, tAgg-def, tDer-def, tComp-def : nécessaires pour représenter le couple trait – définition.

- **Lexique thématique**

Clé d'entrée	Lemme
Informations par clé	Un ou plusieurs chiffres faisant référence aux thèmes du *référentiel thématique*
Format	Texte formaté.
Type	Lexique d'analyse
Exemple	FR : abbaye:443.453

Formalisation :

$$Ls = \{uS_1, uS_2, ..., uS_n\}$$

où $uS = (lem, Dt)$

où $Dt \subset Ts$

$$Ts = \{tS_1, tS_2, ..., tS_n\}$$

Objets :

Dt	description thématique
uS	unité sémantique

Rendu visuel :

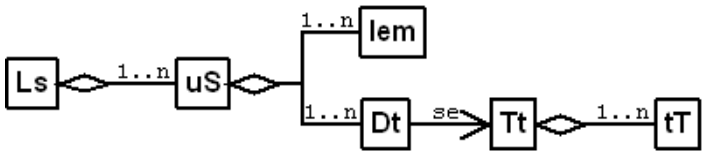


Figure 22 : Diagramme du *Lexique sémantique*

- **Référentiel thématique**

Clé d'entrée	Chiffre
Informations par clé	La présence de la clé est autosuffisante. Elle est accompagnée de l'intitulé du thème, d'exemples, contrexemples, et de codes indiquant la place du thème dans la hiérarchie des thèmes, le thème opposé et l'appartenance à des domaines.
Format	Texte formaté.
Type	Métadonnées
Remarques	Indépendant des langues
Exemple	443 : christianisme, … ; 453 : bâtiment religieux, …

Formalisation :

$$RefS = \{(tT_1, def_1), (tT_2, def_2), ..., (tT_n, def_n)\}$$

Objets :

RefS	Référentiel sémantique
tT	trait thématique

Rendu visuel :

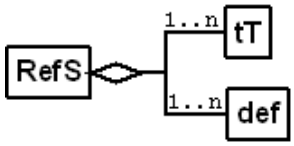


Figure 23 : Diagramme du *Référentiel sémantique*

- Lexique de variantes orthographiques

Clé d'entrée	Unité variante
Informations par clé	Description morphosyntaxique de la variante, mot-forme canonique, description morphosyntaxique du mot-forme canonique
Format	Texte formaté.
Type	Lexique d'analyse
Remarques	Le mot-forme canonique doit être présent comme tel dans le *lexique général* ou *spécifique.* Les variantes sont attachées au même lemme comme s'ils faisaient partie du même paradigme de flexion.
Exemple	FR : acuponcture, nom – acupuncture, nom

Formalisation :

$$Lvar = \{uVar_1, uVar_2, ..., uVar_n\}$$

où $uVar = ((uT_1, cat_1), (uT_2, cat_2), ..., (uT_n, cat_n))$

Objets :

Lvar	Lexique des variantes
uT	unité textuelle (suite de caractères)
uVar	unité de variation

Rendu visuel :

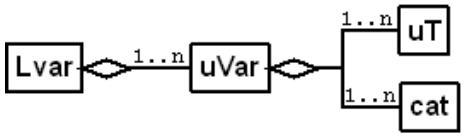


Figure 24 : Diagramme du *Lexique des variantes orthographiques*

- **Lexique d'abréviations et de sigles**

Clé d'entrée	Unité abrégée
Informations par clé	Combinaison du mot-forme abrégé avec les points et sa forme développée.
Format	Texte formaté.
Type	Lexique d'expansion de requête
Remarques	Les formes variantes sans points sont gérées automatiquement.
Exemple	FR : S.D.F. : Sûreté de Fonctionnement

Formalisation :

$$Labrev = \{uAbrev_1, uAbrev_2, .., uAbrev_n\}$$

où $uAbrev = (uT_1, uT_2, ..., uT_n)$

Objets :

Labrev	Lexique des abréviations
uAbrev	unité d'abréviation

Rendu visuel :

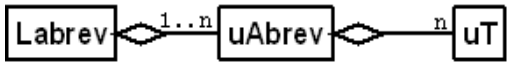


Figure 25 : Diagramme du *Référentiel sémantique*

- **Lexique de nominalisations**

Clé d'entrée	Lemme de verbe
Informations par clé	Une ou plusieurs lemmes de noms qui sont des nominalisations de la clé d'entrée
Format	Texte formaté.
Type	Lexique d'expansion de requête
Remarques	Etablit un lien entre tous les mots-formes des lemmes.
Exemple	FR : nominaliser, nominalisation

Formalisation :

$$Lnom = \{uNom_1, uNom_2, .., uNom_n\}$$

où $uNom = (lem_1, lem_2, ..., lem_n)$

Objets :

Lnom	Lexique des nominalisations
uNom	unité de nominalisation

Rendu visuel :

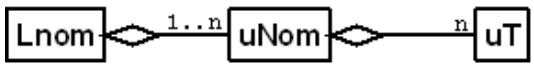


Figure 26 : Diagramme du *Lexique des nominalisations*

- **Lexique de renvoi réciproque**

Clé d'entrée	Lemme
Informations par clé	Un ou plusieurs lemmes qui entretiennent une relation sémantiquement réciproque avec la clé d'entrée
Format	Texte formaté.
Type	Lexique d'expansion de requête
Remarques	Etablit un lien entre tous les mots-formes des lemmes.
Exemple	FR : aride, aridité

Formalisation :

$$Lrrecip = \{uRrecip_1, uRrecip_2, ..., uRrecip_n\}$$

où $uRrecip = (lem_1, lem_2, ..., lem_n)$

Objets :

Lrrecip Lexique des renvois
uRrecip unité de renvoi

Rendu visuel :

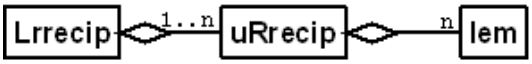


Figure 27 : Diagramme du *Lexique de renvoi réciproque*

- **Lexique de renvois simples**

Clé d'entrée	Lemme
Informations par clé	Un ou plusieurs lemmes dont la signification et la forme sont proches de la clé d'entrée.
Format	Texte formaté
Type	Lexique d'expansion de requête
Remarques	Les lemmes doivent être présents dans le *lexique général* ou *spécifique*. Les liens sont généralement des dérivations autres que nominalisation. Etablit un lien entre tous les mots-formes des lemmes.
Exemple	FR : illustrateur, illustratrice

Formalisation :

$$Lrsimple = \{uRsimple_1, uRsimple_2, ..., uRsimple_n\}$$

où $uRsimple = \{lem_1, rSimple, \{lem_2, lem_3, ..., lem_n\}\}$

Objets :

Lrsimple Lexique des renvois
uRsimple unité de renvoi

Rendu visuel :

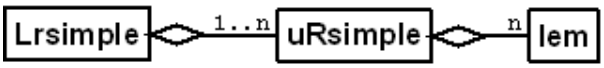


Figure 28 : Diagramme du *Lexique de renvoi simple*

- **Lexique de normalisation**

Clé d'entrée	Unité normalisée
Informations par clé	Une ou plusieurs variantes qui ont le même sens.
Format	Texte formaté
Type	Lexique d'analyse
Remarques	Utilisé comme un lexique de variantes pour les entités. Sert e.a. à la présentation des entités nommées dans les pavés de navigation.
Exemple	Guillem Blancfort ; Blancfort Guillem

Formalisation :

$$Lnorm = \{uNorm_1, uNorm_2, ..., uNorm_n\}$$

où $uNorm = (lem_1, lem_2, ..., lem_n)$

Objets :

Lnorm	Lexique des normalisations
uNorm	unité de normalisation

Rendu visuel :

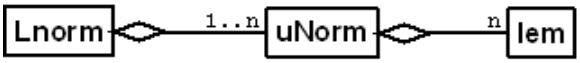


Figure 29 : Diagramme du *Lexique des normalisations*

- **Lexique de n-grammes de lettres**

Clé d'entrée	N-gramme
Informations par clé	Probabilité de la clé.
Format	Texte formaté
Type	Lexique d'analyse
Remarques	Calculé à partir du C*orpus pour calculer les n-grammes*
Exemple	FR : qu 538

Formalisation :

$$Lngram = \{uNgram_1, uNgram_2, ..., uNgram_n\}$$

où $uNgram = \{ngram_1, pNgram\}$

Objets :

Lngram	Lexique de n-gramme
ngram	n-gramme
uNgram	unité de n-gramme
pNgram	probabilité de n-gramme

Rendu visuel :

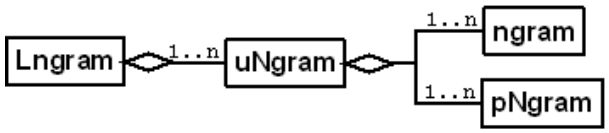


Figure 30 : Diagramme du *Lexique des n-grammes de lettres*

- Lexique de mots fréquents

Clé d'entrée	Mot-forme
Informations par clé	La clé est auto-suffisante.
Format	Texte formaté
Type	Lexique d'analyse
Remarques	La clé existe dans le *lexique général*, et la sélection est faite à partir de données fréquentielles et grammaticales
Exemple	FR : il

Formalisation :

$$Lmfr = \{mfr_1, mfr_2, ..., mfr_n\}$$

où $mfr \in F$

$$F = \{f_1, f_2, ..., f_n\}$$

Objets :

Lmfr	Lexique de mots fréquents
mfr	mot fréquent

Rendu visuel :

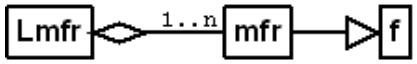


Figure 31 : Diagramme du *Lexique des mots fréquents*

- **Lexique de désambiguïsation**

Clé d'entrée	Unité textuelle du corpus
Informations par clé	Une ou plusieurs catégories grammaticales
Format	Texte formaté
Type	Lexique d'annotation
Remarques	Constitué automatiquement à l'apprentissage du modèle de langage qui contient aussi la *grammaire de désambiguïsation de texte*. Reflète toutes les ambiguïtés présentes dans le *corpus du modèle de langage*.
Exemple	FR : silhouette nom

Formalisation :

$$Ldes = \{uDes_1, uDes_2, ..., uDes_n\}$$

où $uDes = \{uT, Dcat\}$

où $Dcat \subset Cat$

Objets :

Dcat	description catégorielle
Ldes	Lexique de désambiguïsation
uDes	unité de désambiguïsation

Rendu visuel :

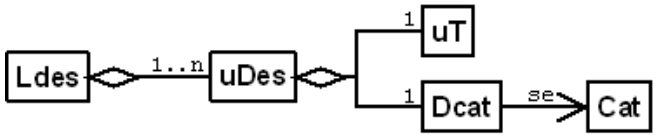


Figure 32 : Diagramme du *Lexique de désambiguïsation*

6.3.2 Corpus

S'il y a de nombreux corpus à Sinequa, seulement les corpus étiquetés servent de ressources au système. Il existe également un corpus de validation pour chaque grammaire décrit dans le paragraphe suivant. Pour des raisons pratiques de communication entre le programmeur et le linguiste, la grammaire et le corpus de validation ont été fusionnés dans un seul fichier. Chaque règle est illustrée par un exemple qui sert en même temps comme exemple de validation du traitement demandé. Les exemples du corpus sont en texte formaté, alors que les descriptions de règle sont pour la plupart en texte libre formaté, qui sera interprété par l'informaticien qui les codera.

- **Corpus du modèle de langage**

Clé d'entrée	Phrase
Informations par clé	Chaque phrase est divisée en unités textuelles accompagnées de leur catégorie grammaticale.
Format	Texte formaté
Remarques	Sert à la création du modèle de langage de désambiguïsation morphosyntaxique et du lexique de désambiguïsation. Si une partie n'a pas été utilisée pour l'entraînement du modèle de langage, elle peut servir à l'évaluation du traitement.

Formalisation :

$$corML = (uPhr_1, uPhr_2, ..., uPhr_n)$$

où $uPhr = ((uT,cat)_1, (uT,cat)_2, ..., (uT,cat)_n)$

où $cat \in Cat$

Objets :

corML corpus à l'origine du modèle de langage
uPhr unité de phrase

Rendu visuel :

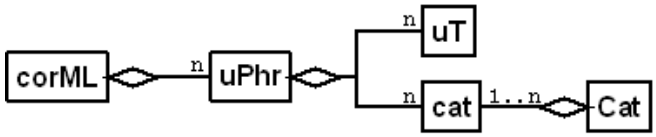


Figure 33 : Diagramme du *Corpus du modèle de langage*

- **Corpus de validation de traitement**

Ces corpus sont des ressources de validation et ne servent pas activement dans le système. Ils sont néanmoins indispensables pour la maintenance des ressources. Dans le cas où les grammaires ne sont pas exprimées de façon totalement formelle, ces fichiers contiennent la description des règles, en texte libre, qui sont codées en dur dans le code de l'analyseur. Le programmeur vérifie la bonne implémentation des règles décrites par le linguiste.

Comme ces corpus ne sont pas des ressources du système, nous n'entrons pas dans le détail de leur description, mais donnons une description générale. Il existe un fichier de validation pour chaque type de traitement. Leur formatage se ressemble énormément, mais s'adapte aux informations du traitement à vérifier.

Clé d'entrée	Exemple de validation
Informations par clé	Description de la règle, n° de la règle, résultat escompté de l'application de la règle
Format	Texte formaté
Remarques	Un fichier par type de traitement (voir les grammaires). Comme on peut déduire de la description, la grammaire et le corpus sont très liés quand la grammaire est exprimée en texte libre.

Formalisation :

$corVal = (uVal_1, uVal_2, .., uVal_n)$

où $uVal = (ex, Dregle, Dval)$

où $Dval$ est le résultat escompté de l'exécution de la règle à valider sur ex.

Objets :

corVal corpus de validation d'un traitement
Dregle description de la règle à valider
Dval résultat de la règle à valider
ex exemple à valider
uVal unité d'exemple annoté

Rendu visuel :

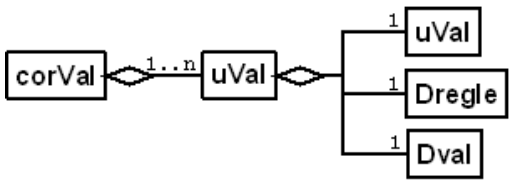


Figure 34 : Diagramme des *Corpus de validation*

- Corpus pour calculer les n-grammes

Clé d'entrée	Suite de caractères
Informations par clé	Nulles : les n-grammes sont calculés à partir du corpus.
Format	HTML
Remarques	Le corpus doit être rigoureusement unilingue.

Formalisation :

$corNgram = (c_1, c_2, .., c_n)$

où c est un caractère

Objets :

corNgram corpus de textes pour le calcul des n-grammes de lettres

Rendu visuel :

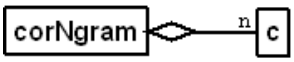


Figure 35 : Diagramme du corpus pour les n-grammes

6.3.3 Grammaires

Nous distinguons deux types de grammaires. Les grammaires lexicales sont des règles lexicales qui servent d'extension au lexique, et les grammaires d'annotation sont des règles d'annotation, c'est-à-dire qui vont enrichir du texte avec des métadonnées. Quand les règles sont représentées en « texte libre », nous l'indiquons par « Métadonnées de règles codées en dur ». Dans ce cas il s'agit d'une grammaire écrite par un linguiste dans un language plus ou moins formel et implémentée par un programmeur.

- **Grammaire d'interprétation**

Clé d'entrée	Etiquette du *référentiel d'étiquettes*
Informations par clé	Règle d'interprétation
Format	Texte libre
Type	Grammaire lexicale. Métadonnées de règles codées en dur
Remarques	Cette grammaire définit les conditions de mise en correspondance de l'unité textuelle et le mot-forme si elles sont différentes des stratégies de mise en correspondance adoptées par défaut.
Exemple	L'équivalence ou non de l'interprétation d'un espace ou d'un tiret dans un même mot composé *: tire bouchon = tire-bouchon*. L'importance de la capitalisation ou des diacritiques pour certains mots : *Mélodie* est nom commun et prénom seulement si le mot est capitalisé.

Formalisation :

$$G\text{int} = \{(tInt, rInt)_1, (tInt, rInt)_2, ..., (tInt, rInt)_n\}$$

où *rInt* décrit la stratégie à suivre pour la mise en correspondance des mots-formes $f \in Lg$ portant les étiquettes concernées *tInt* avec les unités textuelles.

Objets :

rInt règle d'interprétation

Rendu visuel :

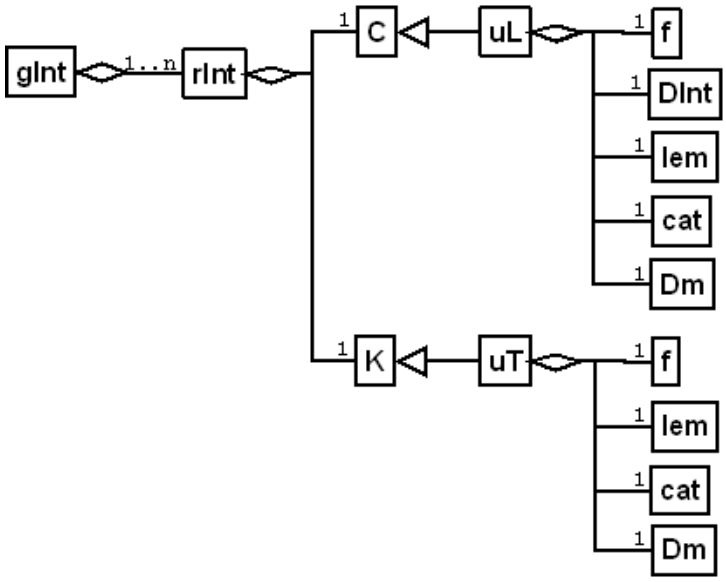


Figure 36 : Diagramme de *Grammaire d'interprétation*

- Grammaire de génération de mots-formes

Clé d'entrée	Un ou plusieurs caractères
Informations par clé	Règle d'interprétation liée à la clé
Format	Texte libre
Type	Grammaire lexicale. Métadonnées de règles codées en dur
Remarques	La présence des caractères de la clé dans un mot-forme déclenche la génération d'autres mots-formes ayant la même description
Exemple	DE : ä > ae, exemple : *Bäcker > Baecker* (boulanger)

Formalisation :

$$Ggen = \{((c_1, c_2, ..., c_n), rGen)_1, ((c_1, c_2, ..., c_n), rGen)_2, ..., ((c_1, c_2, ..., c_n), rGen)_n\}$$

où *rGen* décrit comment générer les mots-formes à partir des mots-formes contenant les caractères référencés pour les ajouter comme *uL* au *Lexique général* et *spécifique*.

Objets :

c	caractère
rGen	règle de génération de mots-formes

Rendu visuel :

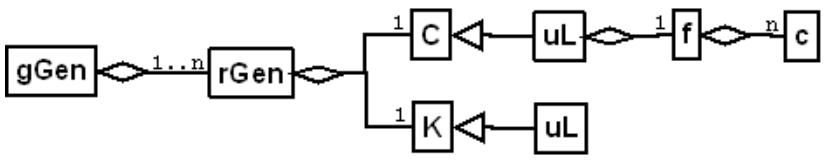


Figure 37 : Diagramme de *Grammaire de génération de mots-formes*

- **Grammaire de décomposition morphologique**

Clé d'entrée	Elément non autonome ou trait descriptif d'élément non autonome.
Informations par clé	Numéro de la règle, description de la règle, exemple de validation
Format	Texte libre
Type	Métadonnées de règles codées en dur
Remarques	S'applique sur les mots hors lexique. La grammaire décrit les combinaisons possibles des éléments non autonomes avec des mots-formes.
Exemple	DE : *Calvin Swifts Humor* > *Swifts* = *Swift* + *s*; FR : *postatomique* (unité textuelle) = *post-atomique* (mot-forme)

Formalisation :

$Gm = \{rM_1, rM_2, ..., rM_n\}$

où rM décrit les combinaisons possibles entre $(e, Dder)$ ou $(e, Dagg)$ et f

Objets :

Gm grammaire de composition morphologique
rM règle de composition morphologique

Rendu visuel :

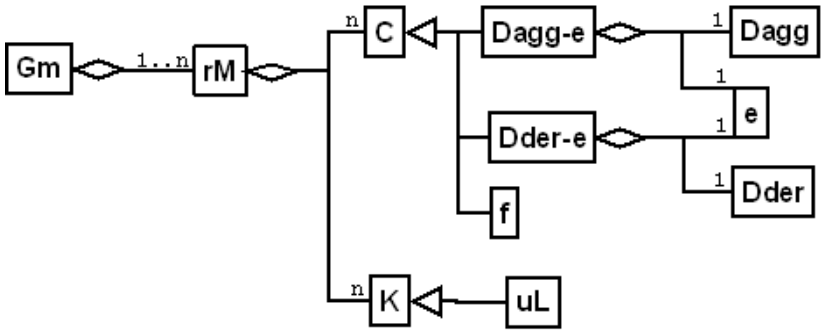


Figure 38 : Diagramme de *Grammaire de décomposition morphologique*

Ajouts de composants structurels pour la visualisation :

Dagg-e, Dder-e : nécessaires pour représenter le couple trait – définition

- **Grammaire de décomposition compositionnelle**

Clé d'entrée	Trait de composition
Informations par clé	Numéro de la règle, description de la règle, exemple de validation
Format	Texte formaté
Type	Métadonnées de règles codées en dur
Remarques	S'applique sur les mots hors lexique et les mots du lexique indiqués comme des mots composés à analyser.
Exemple	NL : *zwemwedstrijd* > *zwem* + *wedstrijd* (concours de natation)

Formalisation :

$$Gcomp = \{(tComp, rComp)_1, (tComp, rComp)_2, ..., (tComp, rComp)_n\}$$

où *rComp* décrit une combinaison possible de (cat, Tm) avec des *eComp* pour former une unité textuelle,

et *tComp* est une référence de règle

Objets :

Gcomp	grammaire de composition
rComp	règle de composition
tComp	référence de règle compositionnelle

Rendu visuel :

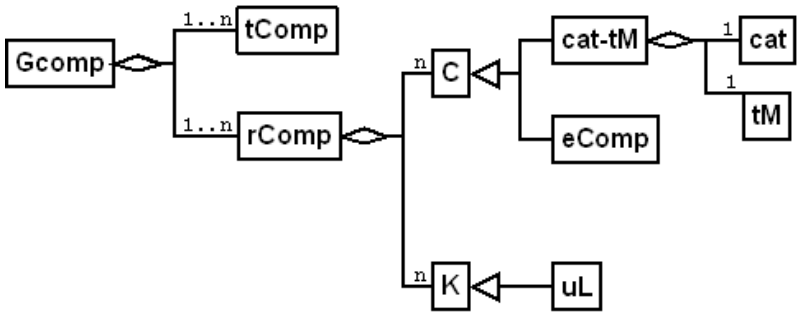


Figure 39 : Diagramme de *Grammaire de décomposition compositionnelle*

- **Grammaire de génération de formes phonétiques**

Clé d'entrée	Une ou plusieurs caractères
Informations par clé	Caractères phonétiques correspondant à la clé, contexte de phonétisation, n° de la règle
Format	Texte formaté
Type	Grammaire lexicale
Remarques	Utilisé pour assouplir la recherche.
Exemple	FR : **.** archéo **.** > aRkeo

Formalisation :

$$Gphon = \{rPhon_1, rPhon_2, ..., rPhon_n\}$$

où $$rPhon = \{(uOrtho, uPhon)_1, (uOrtho, uPhon)_2, ..., (uOrtho, uPhon)_m\}$$

où *uOrtho* est une suite de caractères et *uPhon* est son équivalent phonétique.

Objets :

Gphon	grammaire de phonétisation
rPhon	règle de phonétisation

Rendu visuel :

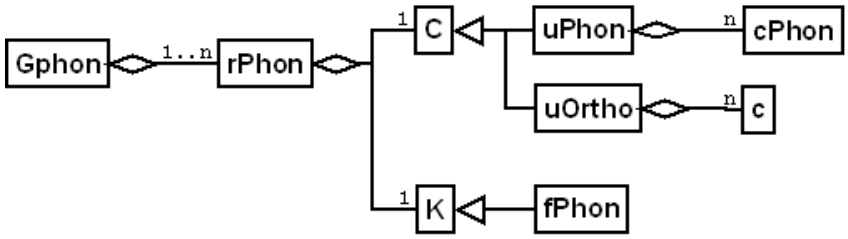


Figure 40 : Diagramme de *Grammaire de phonétisation*.

- Grammaire de classes de flexion

Clé d'entrée	Code de classe de flexion
Informations par clé	Flexions liées à la clé d'entrée, opérations sur la racine.
Format	Texte formaté
Type	Grammaire lexicale
Remarques	Codé en dur ou explicite selon les langues.
Exemple	FR : N5 : + ∅ = nom féminin singulier (ex. grotte) + s = nom féminin pluriel (ex. grottes)

Formalisation :

$$Gflex = \{rFlex_1, rFlex_2, ..., rFlex_n\}$$

où *rFlex* décrit les modalités pour appliquer les flexions d'une classe de flexion *cf* représentée par le code *cFlex* sur un lemme pour obtenir des unités lexicales.

Objets :

cFlex	code de classe de flexion
cf	classe de flexion
pf	paradigme de flexion
rFlex	règle de flexion

Rendu visuel :

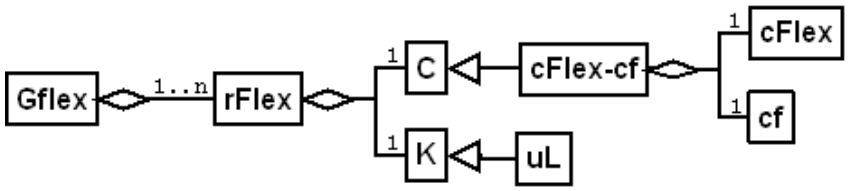


Figure 41 : Diagramme de *Grammaire de classe de flexion*

Ajouts de composants structurels pour la visualisation :

cFlex-cf : nécessaires pour lier la classe de flexion au code la représentant

- Grammaire de désambiguïsation de texte

Clé d'entrée	Suite de catégories grammaticales et/ou d'unités textuelles du corpus désambiguïsation
Informations par clé	Catégorie grammaticale
Format	Texte formaté
Type	Grammaire d'annotation
Remarques	Appris à partir du *corpus du modèle de langage*. Constitue ce modèle avec le *lexique de désambiguïsation*.
Exemple	FR : nom > verbe, -1 = pronom, +1 = le

Formalisation :

$$GdesT = \{rDesT_1, rDesT_2, ..., rDesT_n\}$$

où $$rDesT = (((uT, cat)_1, (uT, cat)_2, ..., (uT, cat)_n), ((uT, cat)_1, (uT, cat)_2, ..., (uT, cat)_n))$$

$$GdesT \cup Ldes = ML$$

Objets :

GdesT	grammaire de désambiguïsation
ML	modèle de langage
rDesT	règle de désambiguïsation de texte

Rendu visuel :

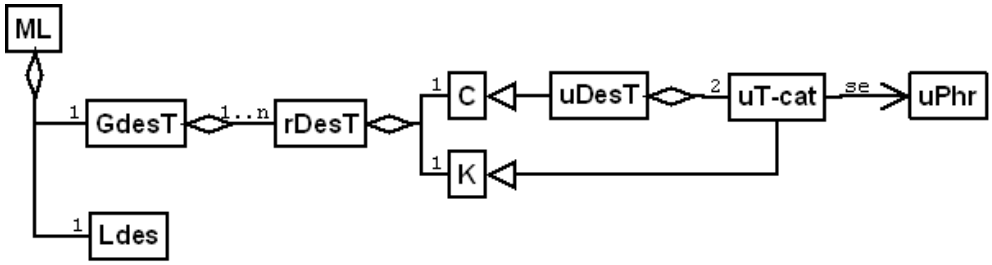


Figure 42 : Diagramme de *Grammaire de désambiguïsation de texte*

Ajouts de composants structurels pour la visualisation :

uT-cat : nécessaires pour représenter le couple trait – définition

- Grammaire de désambiguïsation de requête

Clé d'entrée	Suite de catégories grammaticales
Informations par clé	Contexte, une ou plusieurs catégories grammaticales
Format	Texte libre formalisé
Type	Grammaire d'annotation
Remarques	Codé en dur
Exemple	FR : déterminant&&pronom, + nom > déterminant

Formalisation :

$$GdesR = \{rDesR_1, rDesR_2, ..., rDesR_n\}$$

où *rDesR* décrit comment désambiguïser un mot-forme *f* ambigu selon son contexte.

Objets :

GdesR grammaire de désambiguïsation
rDesR règle de désambiguïsation de requête

Rendu visuel :

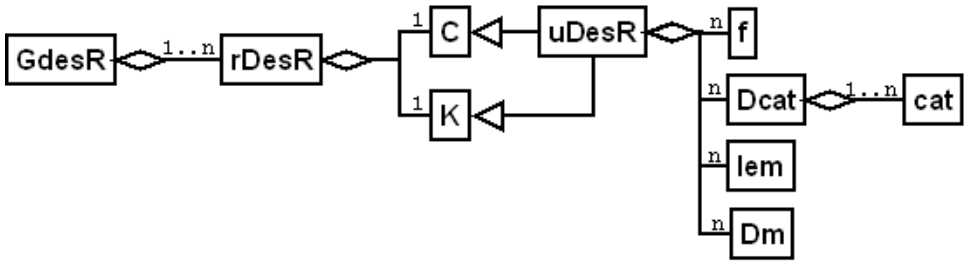


Figure 43 : Diagramme de *Grammaire de désambiguïsation de requête*

- Grammaire de détection d'entités et de relations

Clé d'entrée	Automate
Informations par clé	Type et sous-types d'entité
Format	XML
Type	Grammaire d'annotation
Remarques	Il existe un automate par type d'entité extrait. Ces grammaires incluent des listes de mots-formes et de lemmes qui ressemblent à des lexiques. Ce sont néanmoins des listes de contraintes des règles et non pas des entrées lexicales.
Exemple	

Formalisation :

$$Gent = \{(Aut_1, ent), (Aut_2, ent), ..., (Aut_n, ent)\}$$

où $$ent \in \begin{Bmatrix} groupe\_nominal\_court, nom\_de\_personne, \\ nom\_d'entreprise, nom\_géographique, date, ... \end{Bmatrix}$$

et $$Aut = \{eAut_1, eAut_2, ..., eAut_n\}$$

où $$eAut = \{test, arc\}$$

où $$test \in \{u, f, lem, , cat, tM, tS, Gent, ent\}$$

Objets :

chemin	unité d'automate
ent	type d'entité
Gent	grammaire de détection d'entités
nœud	unité de *chemin*
test	élément minimal de *nœud*

Rendu visuel :

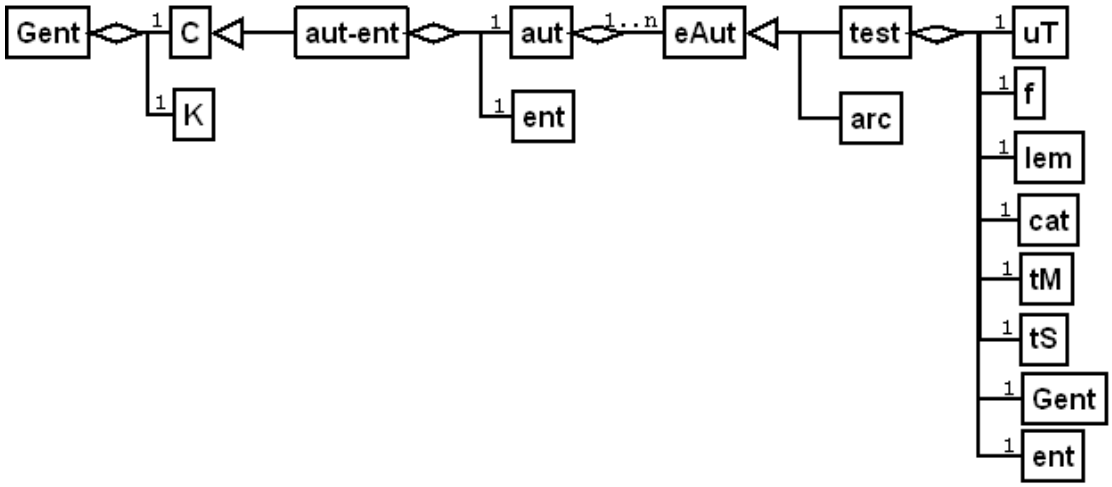


Figure 44 : Diagramme de *Grammaire de détection d'entités*

- Grammaire d'inclusion et de chevauchement d'entités

Clé d'entrée	Deux types d'entité.
Informations par clé	Indication d'inclusion ou de chevauchement.
Format	Texte libre.
Type	Grammaire d'annotation.
Remarques	Codé en dur. Déclenche l'annulation de l'entité incluse.
Exemple	Nom de personne inclus dans lieu : annulation du nom de personne (ex. Bibliothèque François Mittérand).

Formalisation :

$Gincl = \{rIncl_1, rIncl_2, ..., rIncl_n\}$

où $rIncl$ décrit une inclusion de d'entité et l'entité à garder.

Objets :

Gincl grammaire d'inclusion et de chevauchement

Rendu visuel :

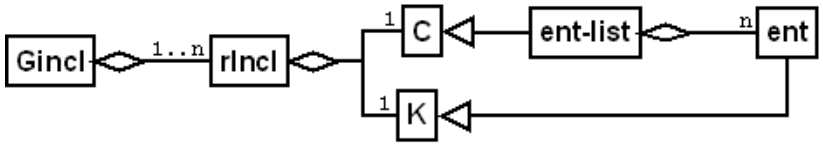


Figure 45 : Diagramme d'inclusion et de chevauchement d'entités

6.4 Conclusion

Avec l'architecture linguistique, nous avons formalisé la façon dont les traitements et les connaissances contenues dans les ressources interagissent. Cette formalisation donne une vision sur le système qui est indispensable si on veut le modifier. Dans le cas de l'ajout d'un traitement, l'architecture permet de vérifier le branchement par la disponibilité des connaissances et de savoir si les nouvelles ressources sont compatibles avec les existantes. En cas de modification structurelle d'une ressource comme la suppression d'un type d'information, on peut vérifier quel traitement sera affecté. Cette architecture peut également révéler des points faibles, comme l'existence de plusieurs ressources contenant les mêmes informations sans qu'elles soient mises en correspondance. Ce genre de duplication est une source d'erreurs fréquente mais ne peut pas toujours être évitée, surtout quand plusieurs acteurs interviennent sur la même plate-forme.

Si une connaissance fine de l'architecture linguistique est également indispensable pour rapidement retrouver la source d'une erreur, nous l'avons surtout utilisée dans nos travaux pour concevoir un outil de mise à jour que nous décrivons dans le chapitre suivant (7.2). Indirectement, nous l'avons également utilisé pour développer les outils de pilotage du chapitre 8 et l'outil d'acquisition du chapitre 9.

En règle générale, aucun travail sur les ressources ou les traitements n'est possible sans une connaissance profonde de l'architecture linguistique. Celle-ci est souvent l'œuvre d'une personne qui en garde le secret, mais ce n'est que par le partage qu'il est possible d'avancer.

PARTIE 4

Création d'un environnement de gestion de ressources

Chapitre 7

Faciliter la mise à jour des ressources

L'environnement de gestion doit faciliter les opérations récurrentes qu'exécutent les linguistes. Dans ce chapitre nous construisons la partie *mise à jour* de l'environnement. Elle est constituée d'un ensemble d'outils qui sont tous indispensables pour une mise à jour cohérente et efficace, même si certains d'entre eux peuvent sembler anecdotiques au premier abord. Chaque outil que nous avons mis en place correspond ainsi à un besoin primaire ressenti par nous-mêmes ou par l'équipe. La première partie de ce chapitre présente les outils de l'environnement. L'un d'entre eux est repris dans la seconde partie de ce chapitre. Il offre un accès unique et global à toutes les ressources lexicales et permet de vérifier les grammaires sur leur validité lexicale. La création de cet outil s'est fortement appuyée sur la modélisation de l'architecture linguistique : il n'aurait pas été possible de le concevoir sans une connaissance profonde des liens entre les informations que contiennent les ressources linguistiques.

7.1 Mise en place d'un environnement de gestion de ressources

7.1.1 Gestion des versions et transfert de données

A notre arrivée à Sinequa en 2002, les ressources linguistiques étaient échangées par répertoire partagé ou par mail. La gestion des versions des ressources linguistiques était totalement entre les mains de l'équipe informatique et lié aux versions du produit. L'introduction d'un système de gestion des versions a permis de sécuriser les ressources et de formaliser les échanges entre linguistes mais aussi avec l'équipe informatique. Techniquement il s'agissait au tout début de CVS[89], qui a été remplacé plus tard par SVN[90]. Habituellement, ce logiciel est utilisé pour le versioning de code source, mais convient également à nos besoins.

SVN a une architecture client-serveur. Le serveur centralise la version de référence d'un répertoire. Sur chaque ordinateur de travail, le client garde la trace des modifications sur le même répertoire en local. Quand la personne envoie ses données modifiées au serveur, la version de référence est mise à jour et devient disponible en téléchargement pour les autres clients. Si plusieurs personnes ont travaillé en même temps sur le même fichier, une fusion est proposée mais doit être manipulée avec précaution. Dans notre cas, il est rare que plusieurs personnes travaillent sur le même fichier en même temps. En général, la répartition est faite par langue : une seule personne travaille sur les données d'une langue donnée, même si certaines langues font exception, notamment l'anglais et le français. Pour éviter des conflits sur les ressources de ces langues, les tâches sont de préférence planifiées de façon intensive sur courte durée. Si les tâches couvrent plusieurs semaines, elles sont découpées en tâches plus courtes bien délimitées. L'amélioration des noms de personnes d'origine chinoise demande par exemple des interventions dans la grammaire, dans les lexiques des prénoms existants, et la constitution d'une liste de noms de famille[91]. L'introduction de SVN a ainsi profondément changé la façon de travailler dans l'équipe linguistique.

7.1.2 Validation des traitements et des ressources

Avec l'équipe nous avons mis en place un cadre de validation des traitements. Il s'agit du composant principal de non-régression linguistique. En validant les traitements, on valide également les ressources.

A chaque fois qu'une ressource est modifiée, la validation du traitement est lancée pour contrôler l'absence de régression. En théorie, la validation doit comprendre un exemple par règle. Nous avons appelé l'ensemble de ces exemples avec l'annotation du résultat attendu des *corpus de validation* (6.3.2). Pour les règles codées dans les automates, l'opération a été intégrée dans l'interface de l'outil d'édition des automates. Elle peut être lancée en cliquant sur un bouton. Les résultats sont affichés ensuite dans une fenêtre surgissante et dans une page HTML.

---

[89] Concurrent Version System : http://www.cvshome.org/
[90] SVN (Subversion): http://subversion.tigris.org/
[91] Les noms de famille chinois constituent une liste restreint d'idéogrammes. A Sinequa, l'utilisation d'une liste de noms de famille pour la détection des noms de personnes est particulière au chinois.

A titre d'exemple, les règles R1 et R2 de désambiguïsation morphosyntaxique peuvent être validées par les exemples respectifs E1 et E2 qui les accompagnent.

R1 La suite d'un pronom personnel suivi d'un mot ambigu déterminant ou pronom personnel suivi d'un mot ambigu nom ou verbe doit être désambiguïsée en pronom – pronom – verbe.

E1 Il la ballade > Il : pronom – la : pronom personnel – ballade : verbe

R2 Le mot suivant « ces » n'est pas un verbe.

E2 Exemple de validation : Ces avions > Ces : déterminant – avions : nom

Les ressources sont exploitées sous forme binaire. Pour être sûr que l'ensemble binaire est cohérent, qu'il contient toutes les ressources nécessaires et qu'il ne présente pas de régression, nous avons mis en place un cadre de vérification, ce qui revient à passer tous les tests sur les ressources avant livraison à l'équipe informatique. Ces tests comprennent la construction de l'ensemble binaire avec les données présentes sur le serveur SVN (et non pas sur la machine locale) en plus de la validation des traitements comme décrite dans 7.1.2 avec ces mêmes données. Cela évite que la livraison du logiciel au client soit bloquée par une intégration infructueuse des ressources linguistiques.

Etant donné que la phase de test demande quelque temps à s'exécuter et qu'il faut prévoir le temps pour corriger d'éventuelles erreurs, l'équipe linguistique a dû changer ses habitudes de livraison pour y inclure ces délais incompressibles de vérification. Cette phase alourdit donc la procédure de livraison, mais elle fluidifie la communication entre les équipes. Techniquement la vérification consiste en un ensemble de scripts mélangeant des commandes DOS et Unix[92] appelant parfois des programmes PERL.

7.1.3 Facilitation de l'utilisation du module d'analyse

Le module qui fournit l'analyse linguistique des documents est hautement paramétrable. L'utilisation courante peut néanmoins être résumée à trois cas d'utilisation : l'analyse de la requête et l'analyse du texte, cette dernière avec ou sans entités nommées. Ce module étant souvent utilisé en mode console, nous avons écrit quelques scripts sous forme de batch pour pouvoir lancer les analyses avec une simple commande. Le module prenant en entrée les ressources sous forme binaire, leur génération est également l'une des opérations les plus courantes, qui peut être aussi lancée en mode batch.

La mise en place de ces batch a nécessité une standardisation de l'environnement de l'utilisateur. Le système d'exploitation Windows offre deux manières d'y arriver : soit en standardisant les variables d'environnement et en laissant libres l'organisation de l'ordinateur, soit en standardisant une partie des répertoires. Nous avons choisi la dernière option pour la raison qu'il nous arrive souvent de prendre en main l'ordinateur d'une autre personne. De cette façon, on retrouve le même environnement, ce qui facilite encore une fois la communication et augmente la productivité.

[92] L'environnement est sous Windows mais nous émulons les commandes de l'environnement Unix dans le shell de Windows grâce à Cygwin (http://www.cygwin.com/).

7.1.4 Mise en place de la documentation linguistique

A notre arrivée à Sinequa, aucune documentation linguistique sur le module d'analyse linguistique n'existait. Nous avons d'abord mis en place une page HTML à partir de quelques notes personnelles. Quand l'équipe s'est agrandie, nous l'avons remplacée par une documentation collaborative sous forme de wiki[93]. Au cours des années elle a été réécrite plusieurs fois pour refléter une structure plus logique par rapport aux informations récoltées. Actuellement elle compte 81 pages avec de nombreux fichiers associés et des liens vers l'extérieur.

Au fur et à mesure que l'équipe s'est agrandie, la documentation est devenue plus difficile à gérer à cause de son esprit collaboratif : ce qui était la grande force pour construire la documentation était devenu sa faiblesse. Chaque nouvelle recrue apportait sa propre vision sur l'organisation interne de la documentation, ce qui a signé l'échec du mode collaboratif. Nous avons donc figé sa structure générale et gardé l'environnement pour sa facilité d'édition.

La documentation des étiquettes morphosyntaxiques et sémantiques, appelée *Référentiel des étiquettes* dans 6.3.1, était gérée de façon non documentée quand nous avons repris la gestion globale des ressources. Un fichier HTML avec un aperçu global mélangeant les étiquettes d'un nombre restreint de langues était généré à partir de fichiers dispersés dans l'arborescence des répertoires partagés. Pour rationaliser la gestion de cette documentation, nous avons repensé son organisation et sa présentation, puisque le nombre de langues avait triplé.

Nous avons fusionné la documentation des étiquettes avec le fichier déclaratif des étiquettes qui est utilisé pour la génération des ressources binaires. De cette manière nous l'avons intégrée dans le répertoire de travail SVN et par conséquent dans la gestion quotidienne des fichiers. La vue globale sur l'ensemble des étiquettes peut être obtenue par simple exécution d'un script qui génère un XML visualisable et navigable grâce à une feuille de style XLS. La navigation est faite par langue, et ensuite par étiquette, en ordre alphabétique.

Dans cette même vue, nous avons ajouté des informations calculées sur les ressources qui contiennent les étiquettes. Il s'agit d'une distinction entre catégories grammaticales et traits morphosyntaxiques indiquant quels traits sont associés à quelles catégories grammaticales, ainsi que la relation inverse. En plus, nous indiquons le nombre de catégories et de traits présents dans les lexiques au moment de la génération de la vue.

7.1.5 Développement d'un accès unique et global aux lexiques

Dans les ressources linguistiques de Sinequa, il existe huit types de lexiques différents, chaque type ayant son propre format d'entrée lexicale selon les informations qu'il contient. Ces types sont illustrés dans la figure 46 qui présente une vue sur les ressources lexicales physiques avec un diagramme de classes en UML[94]. Les couleurs ne font pas partie du modèle standard d'UML, mais indiquent des informations de même type : les mots-formes sont indiqués en vert, les lemmes en jaune, les catégories grammaticales en bleu et les descripteurs sémantiques en lilas. La présence de la même couleur à plusieurs endroits montre que des informations identiques sont physiquement présentes dans plusieurs fichiers.

Cette architecture privilégie la simplicité du codage syntaxique dans plusieurs fichiers à la complexité d'une seule ressource rassemblant toutes les informations. Si elle rend facilement

---

[93] Le wiki en question est le Dokuwiki : http://www.dokuwiki.org/

[94] Pour des raisons de confidentialité, la figure ne représente pas l'architecture la plus récente des ressources lexicales : elle date de fin 2007.

éditables les ressources, elle complique la gestion par le fait que les informations se trouvent dispersées et dupliquées dans des fichiers différents. La simple recherche d'un mot-forme peut ainsi devenir une tâche fastidieuse si on ne sait pas à l'avance dans quel fichier il se trouve, et la duplication d'informations est une source d'erreurs possible.

Pour résoudre ces problèmes d'accès aux informations gérées, nous avons versé toutes les informations dans une base de données centrale dont le modèle relationnel a été inspiré d'une version simplifiée du modèle UML qui a été présenté dans le chapitre 6. Une interface fonctionnellement riche fournit un accès de recherche avancée dans cette base, englobant toutes les informations des ressources ainsi que le référentiel des étiquettes et le référentiel sémantique.

Nous entrons plus en détail sur l'élaboration et les fonctionnalités de ce logiciel dans la section 7.2.

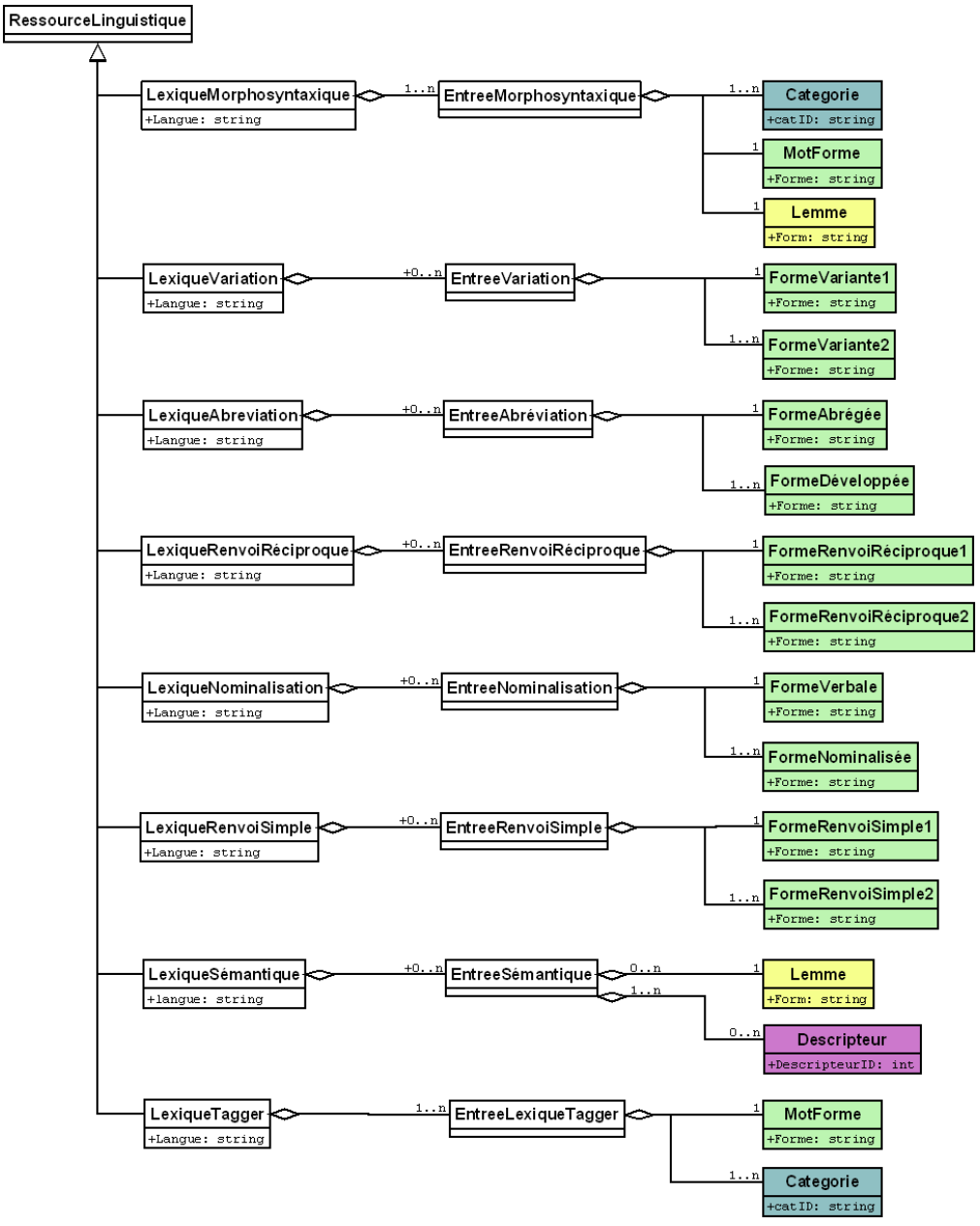


Figure 46 : Diagramme des ressources linguistiques

7.1.6 Centralisation des corpus

Les corpus posent des problèmes de gestion spécifiques qui trouvent leur origine dans le volume de ces ressources. Gourmands en espace disque, les corpus sont compressés au stockage. Nous les avons centralisé et rationalisé l'arborescence qui les contient. Le critère de classement est la langue, ce qui est un classement peu satisfaisant, car nous avons des corpus multilingues pour lesquels nous avons dû créer une autre branche au même niveau.

Comme les fichiers compressés sont très longs à décompresser en cas de simple exploration, il est préférable de mettre en place un système de métadonnées indépendant des corpus qui décrit leur contenu. Nous avons listé les informations nécessaires pour une description complète dans l'Annexe I.

Il existe deux normes de méta-données exploitables pour la description de corpus : OLAC et IMDI[95]. Si les deux peuvent s'adapter à nos besoins, nos préférences vont quand même vers IMDI pour la raison que OLAC nous semble trop ancré dans une approche descriptive de la langue. A titre d'illustration, voici les définitions des 3 valeurs possibles de l'attribut « type de données linguistique »[96] selon OLAC :

- *lexicon: The resource includes a systematic listing of lexical items.*
- *primary_text: Linguistic material which is itself the object of study, typically material in the subject language which is a performance of a speech event, or the written analog of such an event.*
- *language_description: The resource describes a language or some aspect(s) of a language via a systematic documentation of linguistic structures.*

Ces définitions ne demandent aucune formalisation des ressources et les valeurs ne sont pas exclusives : si un lexique avec des définitions (*lexicon*) accompagne un texte (*primary_text*), les deux ressources reçoivent le double étiquetage.

Cette prise de position descriptive d'OLAC n'est pas étonnante puisque SIL International, très engagé dans la description de langues en danger[97], est force motrice dans OLAC. Elle l'utilise notamment pour donner un accès aux ressources langagières des langues en danger qu'elle étudie. L'approche de IMDI est plus générique et pour cela plus adaptée à nos besoins. A titre de comparaison, voici les valeurs possibles pour une « ressource textuelle » selon IMDI ([IMDI, 2003]) :

- *Primary Text: Linguistic material which is the object of study.*
- *Annotation: An annotation of the linguistic material under study.*
- *Lexical analysis: A lexical analysis of the linguistic material under study.*
- *Ethnography [non défini]*
- *Study: The written resource is used for a specific subfield of linguistic science. (This type should be used to select values from the OLAC Linguistic Subject Vocabulary [OLAC-LSV].)*

Il est ensuite possible de spécifier un sous-type, avec par exemple une des valeurs suivantes pour *Lexical analysis* : *dictionary*, *terminology*, *wordlist*, *lexicon*.

---

[95] Voir 4.1.2 pour plus d'informations sur les origines d'IMDI et d'OLAC
[96] *Linguistic Data Type* : http://www.language-archives.org/REC/type.html
[97] A des fins parfois peu scientifiques : [Calvet, 1987], [Manfredi, 2007], [Gallaher, 2007].

Avec le développement d'IMDI, le MPI a mis à disposition sur leur site web un ensemble d'outils pour l'édition, l'indexation et le partage. Si nous avions encore des doutes, cela nous aurait fait trancher en faveur d'IMDI.

Le projet INTERA (*Integrated European language data Repository Area*) réalise les objectifs initiaux d'ISLE : il exploite le format IMDI et rassemble de façon décentralisée les métadonnées de ressources afin de les partager sur internet et de les rendre interrogeables et navigables à travers le IMDI-browser. L'architecture est illustrée dans [Broeder et al., 2004]. A cause de la confidentialité des données, Sinequa ne pourrait pas s'intégrer dans ce réseau, même si l'accès aux sources peut être restreint.

Nous avons donc proposé de mettre en place un système de métadonnées IMDI pour les corpus. Son déploiement n'étant cependant pas prioritaire, cela n'a pas été réalisé. Certaines informations sont aujourd'hui difficiles à retrouver même si elles sont primordiales, comme par exemple la *date de péremption* du corpus qui est liée aux conditions de prêt imposées par le client, le partenaire ou le vendeur.

7.2 Un accès unique et global aux ressources lexicales

7.2.1 Analyse des besoins

Comme indiqué dans la section 7.1.5, les informations se trouvent dispersées dans un grand nombre de fichiers, ce qui rend difficile leur gestion. Pour éviter la perte de temps liée à la recherche d'informations dans les lexiques, nous avons défini le cahier des charges suivant pour le prototype que nous avons mis en place.

Le prototype fournit un accès unique et global aux lexiques, c'est-à-dire que les lexiques et toutes les toutes les informations les concernant sont accessibles depuis la même interface de façon intégrée. La documentation des codes morphosyntaxiques et sémantiques sont par exemple à portée de clic des catégories mentionnées. Les informations qui sont exploitées par des règles indiquent cette liaison afin de préserver la cohérence des règles quand les ressources lexicales sont retravaillées.

L'accès aux lexiques se fait sous forme de filtre, et tous les types d'informations peuvent servir de critère de filtre. Plusieurs critères peuvent être combinés pour exécuter des filtrages avancés. Les expressions régulières sont le moyen de prédilection pour exprimer les filtres. Des filtrages plus compliqués peuvent se faire en SQL. Le but étant de gagner du temps lors de la gestion des données, une attention spéciale est portée à l'ergonomie des opérations les plus fréquentes. Les données filtrées peuvent s'exporter dans le format spécifié par l'utilisateur.

Les performances sont essentielles : les filtrages doivent fournir une réponse immédiatement. Le temps de chargement de données mises à jour doit être raisonnable : il ne doit pas dépasser la demi-journée.

Les besoins suivants faisaient partie de notre réflexion initiale mais n'ont pas été jugés comme essentiels pour Sinequa : mention de l'auteur, date et type de modification et mécanisme de validation des modifications apportées. Ces métadonnées n'existent pas[98] dans les données actuelles. Elles sont indispensables pour une équipe d'édition avec une édition multi-utilisateur et des vérificateurs des entrées, ce qui n'est pas le cas à Sinequa.

[98] Nous les avons indirectement introduits avec la mise en place du système de suivi des versions (7.1.1).

7.2.2 Architecture et choix techniques

Dans la littérature scientifique, plusieurs solutions ont été proposées pour implémenter une base de connaissances lexicales (LKB[99]). Dans leur travail sur l'Oxford Electronic Dictionary, [Gonnet et Tompa, 1987] argumente que les bases de données ne sont pas une bonne solution pour contenir du texte structuré. A la fin des années 90, les technologies ayant évoluées, [Tompa, 1997] propose le texte comme un type de données à part entière avec des opérateurs qui lui sont propres. Nous ne sommes néanmoins pas exactement dans le cas décrit qui est celui d'un dictionnaire électronique contenant de grands morceaux de texte à titre d'exemple. Les travaux de [Neuhaus, 1986] montrent que des bases de données ont été utilisées avec succès pour encoder du lexique dès leur apparition : cet auteur a mis en place une base de données contenant un lexique historique anglais et le lexique des œuvres de Shakespeare dès 1983.

Les expériences décrites dans [Wittenburg et al., 2004] sont plus proches de notre réalité. L'utilisation d'une base de données pour la création du lexique Celex a été couronnée de succès en dépit des réticences initiales des linguistes travaillant sur le lexique. Le modèle relationnel se prête à la description linguistique et l'utilisation d'une base de données était la seule façon de garantir l'intégrité des données dans un contexte hautement multi-utilisateur. Les mêmes auteurs font une comparaison entre une base de données et XML, mais l'absence d'une solution de base de données XML ayant fait ses preuves joue en sa défaveur pour l'utilisation que nous visons.

La verbosité de XML est un frein à son utilisation. La mise sous format XML des lexiques entraînerait la multiplication d'un facteur trois au minimum de la taille des fichiers. Or, certains des lexiques de Sinequa font déjà plusieurs centaines de méga-octets en texte formaté. Alors que leur chargement dans un éditeur de texte n'est déjà pas évident, ils ne seraient plus éditables tels quels en XML. Mêmes avec des technologies spécialisées, l'utilisation native de XML pose des problèmes de performance. La transformation des données actuelles n'étant pas à l'ordre du jour, le choix se pose entre construire la structure de données en mémoire ou bien verser tout dans une base de données comme dans Celex. Nous avons opté pour cette dernière option pour plusieurs raisons. La robustesse des bases de données face à la masse de données à laquelle nous sommes confronté est un facteur primordial. Un autre point fort est que les procédures d'accès simultané par plusieurs utilisateurs sont prévues en standard.

L'architecture globale que nous avons choisie est celle d'un logiciel client avec une interface graphique interrogeant une base de données distante (figure 47). Le choix du langage s'est porté sur C#. Ce choix nous enferme dans le monde propriétaire de Microsoft, mais celui-ci est l'environnement standard de travail à Sinequa. Si nécessaire, une utilisation sous Linux pourrait reposer sur Mono[100], l'alternative open source qui implémente les standards C# (ECMA-334) et CLI[101] (ECMA-335) pour des plateformes basées sur UNIX. La principale raison à l'origine de ce choix est la compatibilité avec le produit Sinequa CS, dont certaines couches reposent sur le socle .NET. Nos développements peuvent ainsi être directement mis à contribution dans de nouvelles fonctionnalités du produit de Sinequa. Nous avons aussi voulu rester dans le cadre logiciel défini par l'entreprise plutôt que d'y faire exception. Le choix du

---

[99] Appelé en anglais *Lexical Knowledge Base* (LKB)
[100] Voir http://www.mono-project.com/.
[101] Common Language Infrastructure (CLI). Plus d'information sur la normalisation de C# et CLI se trouve sur le site suivant : http://www.dotnetexperts.com/ecma/.

SGBD s'est alors porté sur Microsoft SQL Server 2005 (9.0.3042), pour la seule raison de la bonne intégration avec l'environnement choisi.

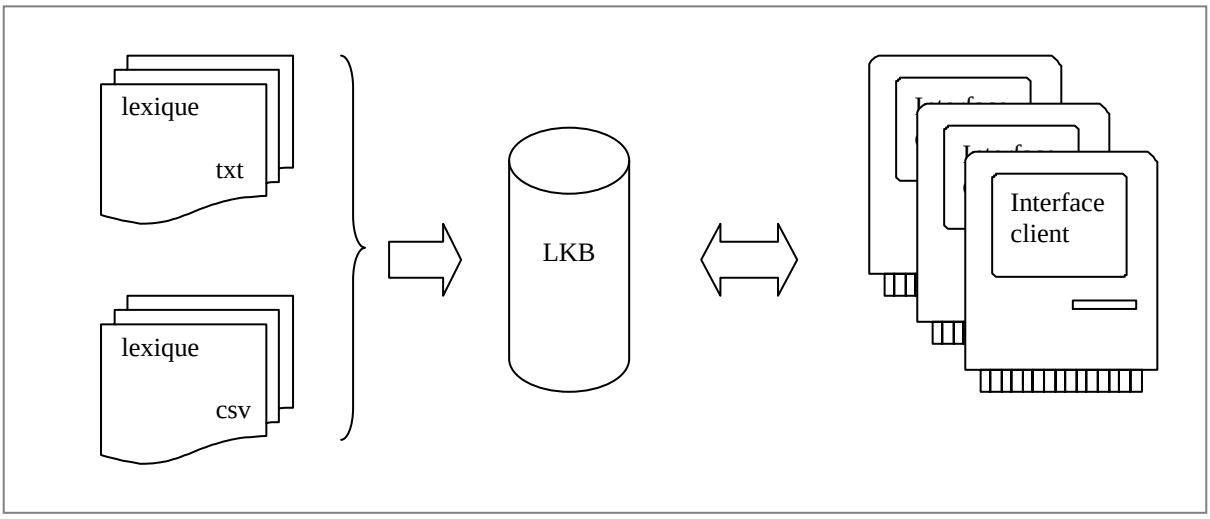


Figure 47 : Architecture informatique

Les bases de connaissances lexicales (LKB) pour quatre langues ont été installées sur un serveur de test linguistique abritant un processeur Dual-Core AMD Opteron Processor 2220 SE 2,80 Ghz avec 8 Gb de RAM sous Windows Server 2003. Les ordinateurs de bureau sur lesquels sont exécutés les clients sont minimalement dotés d'un processeur à 2,4 Ghz avec 2 Go de mémoire vive sous Windows XP Pro. Même si le SGBD a plutôt tendance à prendre beaucoup de ressources coté serveur, aucun ralentissement dû à notre application n'a été constaté ni sur le serveur, ni sur les ordinateurs de bureau. Si nous voulions mettre toutes les langues, il faudrait sans doute un serveur dédié afin de préserver les performances.

7.2.3 Mise en place de la base de connaissances lexicales

Avant de développer le modèle relationnel de la base de connaissances lexicales, nous avons d'abord tenu à expliciter les informations concernées. En partant du modèle ensembliste présenté en 6.3, nous avons visualisé ces informations dans un diagramme UML. La figure 48 montre la version de développement qui a servi à expliciter les liens entre les différentes informations concernées : elle contient les informations des lexiques et celles des grammaires qui sont en interaction directe avec les ressources lexicales. Ce diagramme ne contient pas de conteneurs de classes, comme les classes *grammaire* ou *lexique*. Les couleurs ne font pas partie de la modélisation UML standard, mais nous les avons ajoutées pour découper le schéma en parties liées selon la couche de traitements. Ainsi les classes des grammaires sont indiquées en jaune, les objets verts sont des objets morphosyntaxiques et les objets liés à la sémantique de surface sont lavande.

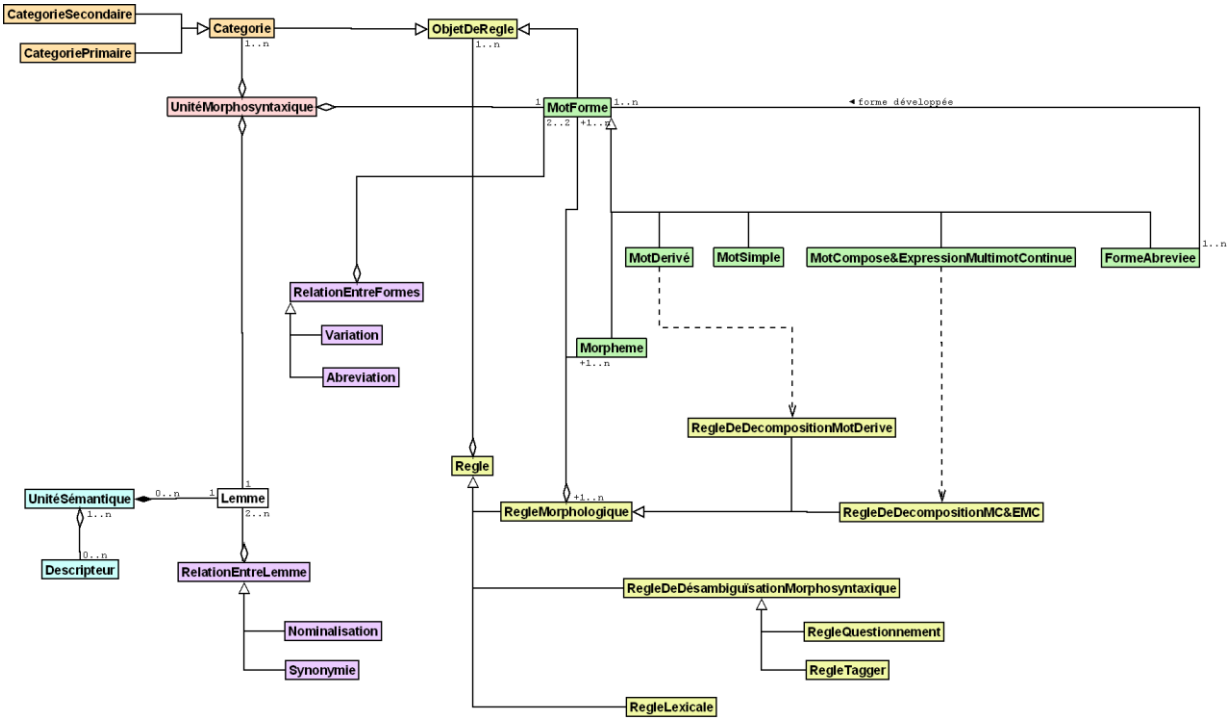


Figure 48 : Le diagramme UML des informations lexicales

Une fois les relations entre les informations tirées au clair, nous avons créé le modèle relationnel. Ce modèle est le résultat d'un processus de test et de réflexion pour trouver un modèle efficace et adapté à nos besoins.

La figure 49 montre le schéma relationnel final[102]. Il comporte 9 tables, dont 3 tables de jointure, c'est-à-dire des tables qui servent à exprimer des relations « plusieurs-à-plusieurs » par l'utilisation de clés étrangères.

[102] Trois schémas antérieurs peuvent être consultés dans l'Annexe D.

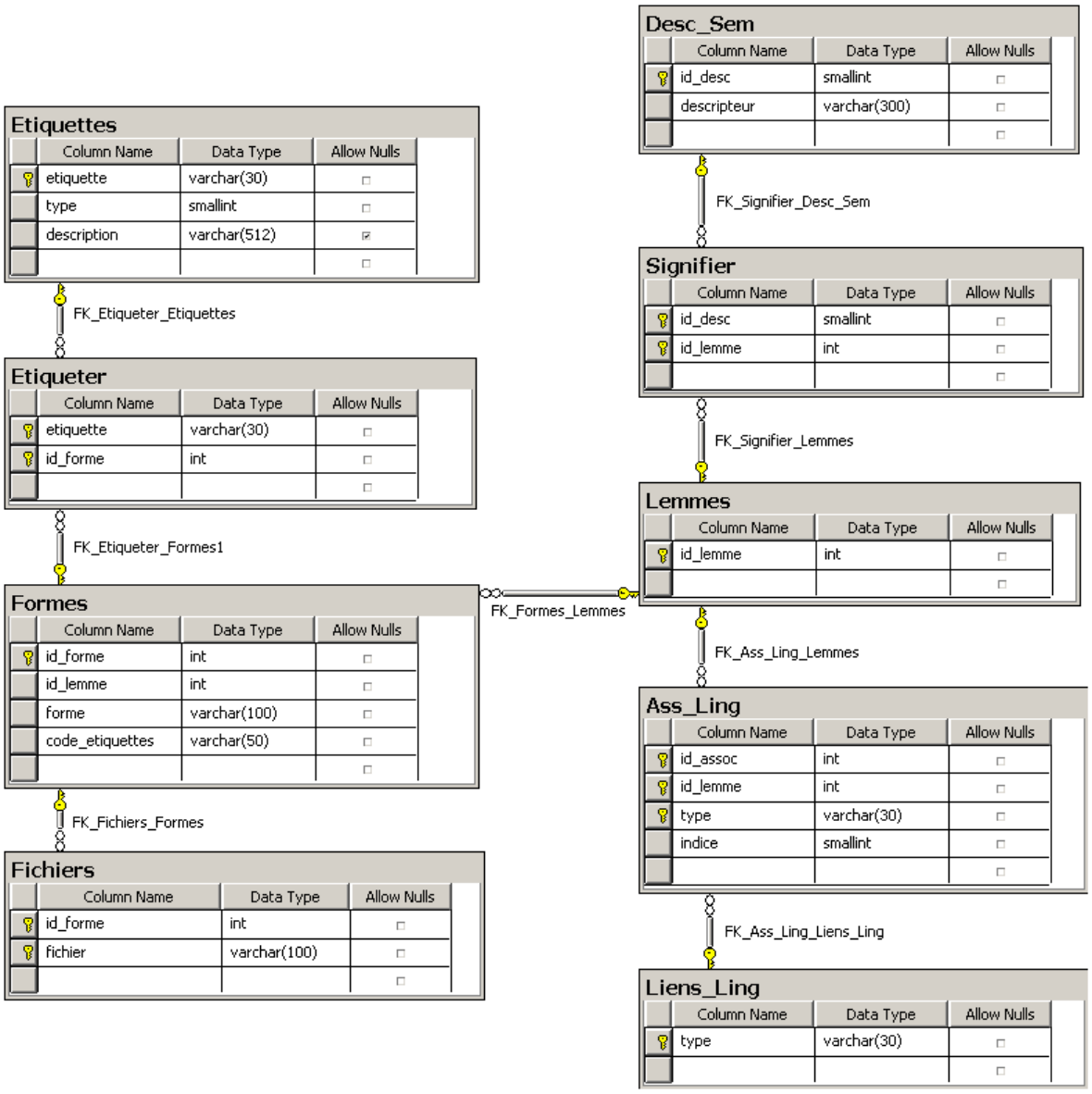


Figure 49 : Schéma relationnel de la base de données

La table « Formes » est la table centrale du modèle. Elle contient le mot-forme, son identifiant, et l'identifiant du lemme associé. Le mot-forme n'est pas unique, contrairement à son identifiant. Si un mot-forme est ambigu, il apparaîtra donc dans plusieurs enregistrements. Les identifiants des formes qui sont des lemmes sont listés dans l'entité *Lemmes*. Si le lemme est ambigu, le bon identifiant de forme est choisi grâce à la catégorie grammaticale.

A titre d'exemple, on voit dans la figure 50 que le mot-forme *manger* est ambigu comme nom et comme verbe. L'identifiant de ce même mot-forme figure ensuite dans le champ de lemme des formes fléchies. Ainsi l'identifiant de lemme de *mangeais* est 197711, ce qui nous ramène vers le verbe à l'infinitif *manger*. La colonne « code_etiquettes » n'a qu'une valeur informative, les valeurs réelles étant codés dans la table de jointure « Etiqueter ».

	id_forme	id_lemme	forme	code_etiquettes
	197710	197710	mangeotter	IN.VER
	197711	197711	manger	IN.VER
	197712	197712	manger	M.NOM.S
	197713	197713	mangerie	F.NOM.S
	660148	197711	mangeais	II.JE.VER
	660149	197711	mangeais	II.TU.VER
	660150	197711	mangeait	II.IL.VER

Figure 50 : Extrait de la table « Formes »

La provenance de chaque entrée est marquée par la table « Fichiers » qui associe un nom de fichier à chaque identifiant de mot-forme afin qu'on puisse soit retrouver facilement le fichier physique qui contient le mot-forme soit filtrer sur cette information.

L'entité « Etiquettes » contient toutes les étiquettes morphosyntaxiques et sémantiques avec leur définition. La figure 51 montre un exemple où on voit les étiquettes POS, PPR et PPS avec le début de leur définition. Une table de jointure, appelé « Etiqueter », lie les identifiants de mots-formes avec ces étiquettes. Le type fait référence au rang de l'étiquette, primaire (= catégorie grammaticale), secondaire (= trait morphosyntaxique) ou tertiaire (trait sémantique).

	etiquette	type	description
	POS	2	Possessif (ex: sien,...
	PPR	2	Participe présent ...
	PPS	1	Participe passé ...

Figure 51 : Extrait de la table « Etiquettes »

De la même façon que « Etiquettes » contient les définitions des étiquettes morphosyntaxiques et sémantiques, « Desc_sem » contient les définitions des descripteurs thématiques. Ces descripteurs sont décrits par des chiffres de 1 à 810 et sont associés aux lemmes par la table de jointure « Signifier ». La figure 52 montre un exemple pour les descripteurs 310 à 312, avec leur définition.

	id_desc	descripteur
	310	tissu GEN 780 chair humaine EX adipeux, gangrè...
	311	manger, aliment SPS 748 (repas) 749 (plat) 365 (...
	312	digestion EX bile, estomac, cuver, aigreur, bourr...

Figure 52 : Extrait de la table « Desc_sem »

L'entité « Lien_ling » (figure 53) contient les 4 types d'associations qui existent dans nos lexiques et qui représentent des renvois simples ou doubles. Les types sont exploités dans la table de jointure « Ass_ling » où ils sont associés à deux ou plusieurs id de lemme. Ces entités sont illustrées dans la figure 54 et la figure 55. *See Also* désigne un renvoi simple. L'indice dans « Ass_ling » indique la direction du lien. Les exemples ci-dessous illustrent le lien de

nominalisation entre abasourdir et abasourdissement. Leurs identifiants de lemme sont liés dans « Ass_ling ».

	type
▶	Acronyme
	Nominalisation
	See Also
	Variante

Figure 53 : La table « Lien_ling »

	id_assoc	id_lemme	type	indice
	2	464	Nominalisation	0
	2	466	Nominalisation	1
	2	565	See Also	0
	2	566	See Also	1

Figure 54 : La table « Ass_ling »

	id_forme	id_lemme	forme	code_etiquettes
	464	464	abasourdir	IN.VER
	465	465	abasourdissant	ADJ.M.S
	466	466	abasourdissement	M.NOM.S

Figure 55 : La table « Formes »

Les quatre liens linguistiques sont bien ceux proposés dans le diagramme UML avec la différence que les liens entre lemmes ou entre formes ne sont pas distingués. En effet, ils ont tous été implémentés comme des liens entre lemmes. Selon l'exploitation faite de ces informations, il n'y a pas de réelle différence entre les deux puisque les mots-formes sont pris en compte comme des lemmes. Dans l'absolu, il est vrai que les équivalences de variation orthographique et d'acronymie se situent au niveau des formes, mais cela revient en pratique au même si on les considère au niveau de la forme ou au niveau de lemme. Ainsi l'ensemble des équivalences *clé / clef* et *clés / clefs* égale l'équivalence entre lemmes *clé / clef*, sachant qu'il existe les lemmatisations *clé* pour *clé*/*clés* et *clef* pour *clef*/*clefs*.

Pour l'instant, aucune base pour une langue compositionnelle n'a été mise en place. Si c'était le cas, il faudrait complexifier légèrement le modèle pour ajouter les composants des mots composés dont la décomposition est codée dans le lexique. Cela ne concerne que quelques mots composés qui ne sont pas couverts par la grammaire de décomposition ou des mots composés dont la décomposition par les règles est ambiguë et erronée. Quelques exemples en allemand de mots composés non couverts par la règle sont donnés en 8.3.2.2c et concernent la composition avec des mots trop courts pour être pris en compte par les règles générales, comme les mots *Ei* et *Öl*.

La question de savoir comment nous pouvions intégrer les règles dans la base de connaissances lexicales était particulièrement difficile à résoudre. Les automates étant des structures complexes en XML, leur ajout dans base de connaissances lexicales nous a paru absurde. Le besoin qui existe est de savoir pour chaque mot-forme s'il existe une règle

associée à ce mot-forme, pour éviter que la modification ou la suppression d'un mot-forme ne rende la règle caduque. A défaut d'accrocher des références de règles à des entrées lexicales, nous avons mis en place une solution qui allège le modèle plutôt que de le complexifier. Les règles ne sont pas chargées dans la base de connaissances lexicales, mais nous avons introduit la possibilité de vérifier l'existence des connaissances utilisées dans les règles dans la base de connaissances lexicales. Cette vérification repose sur le client et sera abordée dans le point suivant (7.2.4).

7.2.4 Fonctionnalités et interface

La conception de l'interface et de ses fonctionnalités sont le fruit de notre collaboration avec le stagiaire que nous avons encadré pendant trois. Nous ne présentons que brièvement les fonctionnalités de l'interface. Plus de détails peuvent être trouvés dans le rapport de stage [Guerra, 2006] qui contient le compte rendu, le manuel de l'interface et une API complète en pseudo-code. Cet API est constituée d'une soixantaine de pages et décrit toutes les opérations possibles sur la base de données.

L'interface permet trois types d'interactions entre l'utilisateur et la base de connaissances lexicales (LKB) : la consultation, l'exportation et la validation. Ces interactions sont illustrées dans la figure 56.

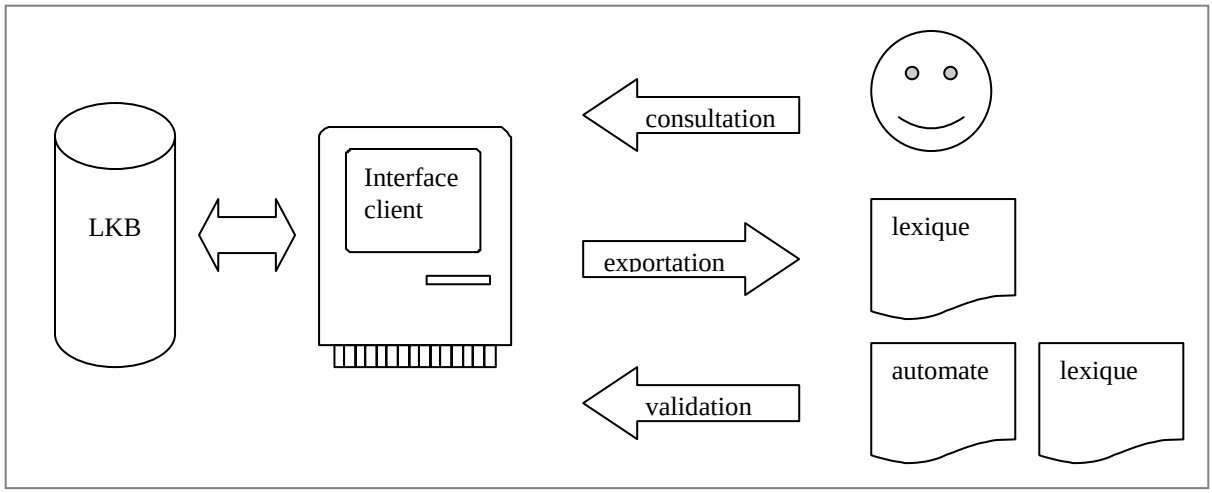


Figure 56 : Interactions avec l'utilisateur

a. Fonctions de consultation

Une fenêtre permet à l'utilisateur de spécifier les critères de filtrage. Les filtres peuvent être mono- ou multicritères et sont listés dans la figure 57. Le filtre se transforme alors en requête SQL qu'il est possible de visualiser et de modifier dans une fenêtre d'édition libre.

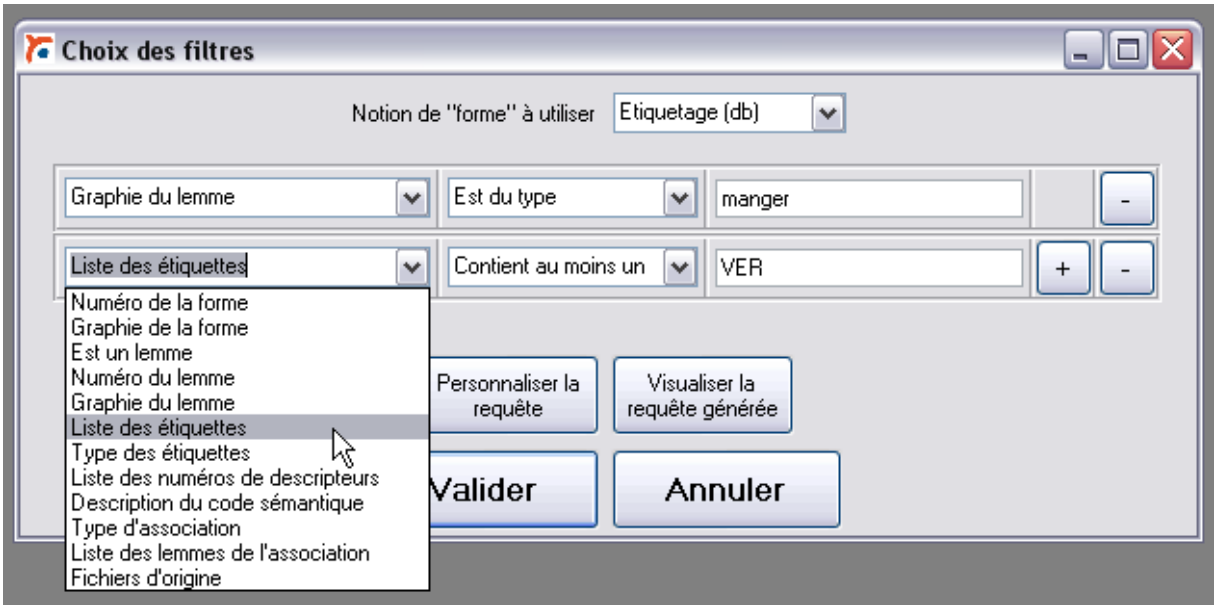
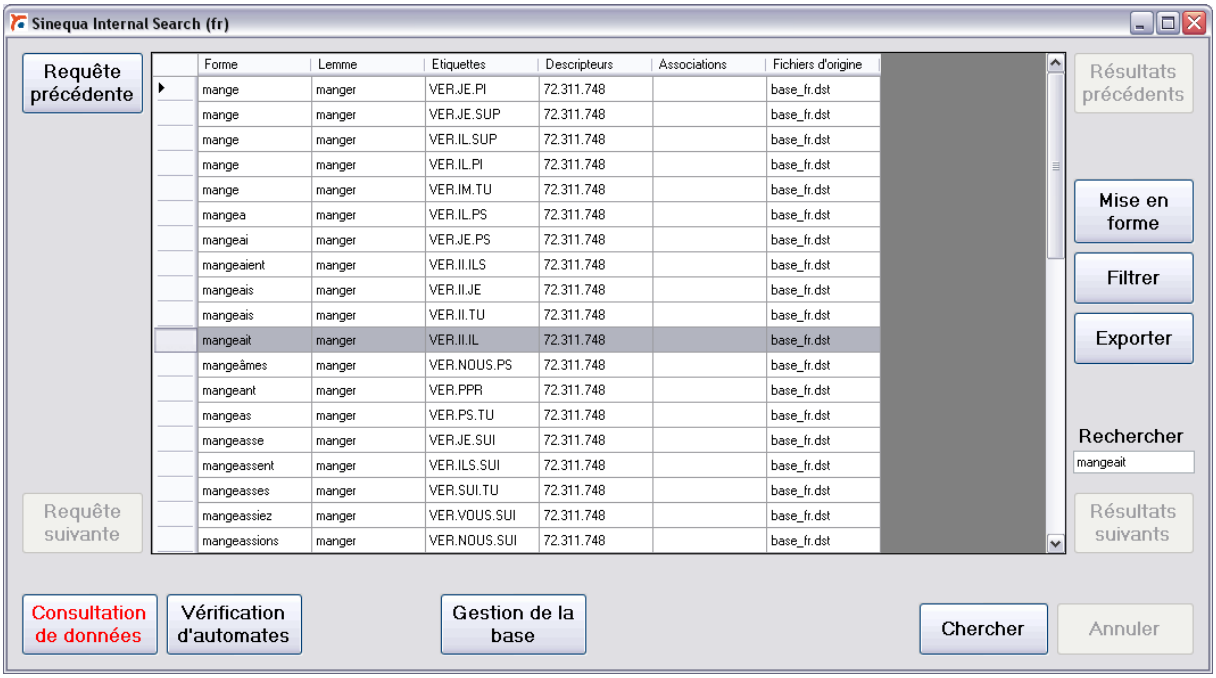


Figure 57 : Fenêtre d'édition de filtre : on requête le lemme *manger* comme verbe.

L'affichage des résultats est proche de la visualisation de l'éditeur de texte mais gagne en clarté puisqu'il est structuré : il y a des colonnes différentes pour le mot-forme, le lemme et les différents types d'informations. La figure 58 montre le résultat de la requête illustrée en figure 57 qui sélectionne le lemme *manger* et le restreint au verbe. L'interface prévoit un champ de recherche qui s'exécute sur toutes les informations affichées, ce qui permet de retrouver toutes les entrées correspondant à la requête. Sur la copie d'écran de la figure 58 la requête est *mangeait*.

Si on garde le pointeur de la souris sur les colonnes Etiquettes et Descripteurs, une fenêtre surgissante montre la documentation associée aux codes utilisés. Une copie d'écran dans l'Annexe E (figure 87) illustre cet affichage pour les informations sémantiques (colonne « Descripteurs ») du mot *avocat*. Cette même copie d'écran affiche son équivalent féminin *avocate* qui lui est associé avec une relation *See Also* dans la colonne « Associations ».



Forme	Lemme	Etiquettes	Descripteurs	Associations	Fichiers d'origine
mange	manger	VER.JE.PI	72.311.748		base_fr.dst
mange	manger	VER.JE.SUP	72.311.748		base_fr.dst
mange	manger	VER.IL.SUP	72.311.748		base_fr.dst
mange	manger	VER.IL.PI	72.311.748		base_fr.dst
mange	manger	VER.IM.TU	72.311.748		base_fr.dst
mangea	manger	VER.IL.PS	72.311.748		base_fr.dst
mangeai	manger	VER.JE.PS	72.311.748		base_fr.dst
mangeaient	manger	VER.II.ILS	72.311.748		base_fr.dst
mangeais	manger	VER.II.JE	72.311.748		base_fr.dst
mangeais	manger	VER.II.TU	72.311.748		base_fr.dst
mangeait	manger	VER.II.IL	72.311.748		base_fr.dst
mangeâmes	manger	VER.NOUS.PS	72.311.748		base_fr.dst
mangeant	manger	VER.PPR	72.311.748		base_fr.dst
mangeas	manger	VER.PS.TU	72.311.748		base_fr.dst
mangeasse	manger	VER.JE.SUI	72.311.748		base_fr.dst
mangeassent	manger	VER.ILS.SUI	72.311.748		base_fr.dst
mangeasses	manger	VER.SUI.TU	72.311.748		base_fr.dst
mangeassiez	manger	VER.VOUS.SUI	72.311.748		base_fr.dst
mangeassions	manger	VER.NOUS.SUI	72.311.748		base_fr.dst

Figure 58 : Fenêtre des résultats

Avec un minimum de connaissances en SQL, l'extraction d'un ensemble de mots répondant à plusieurs contraintes devient un jeu d'enfant alors qu'en comparaison l'écriture d'un script aurait été relativement longue. Il est ainsi possible d'écrire un filtre pour extraire tous les mots qui sont codés uniquement au singulier ou ceux uniquement au pluriel. Si ce codage est justifié dans un nombre restreint de cas, ils peuvent aussi être indicateurs d'un manque ou d'une erreur.

Les filtres ne doivent pas forcément être compliqués pour être utiles. Ainsi, la sélection de toutes les formes d'un même lemme est une fonctionnalité précieuse qui demandait auparavant de jongler avec un script et des fichiers de sortie.

b. Fonction d'exportation

Une fenêtre permet de sélectionner les informations et de les mettre en forme selon les souhaits de l'utilisateur. Une copie d'écran illustre cette fenêtre d'édition de la mise en forme en Annexe E (figure 88).

c. Fonctions de validation

Une première validation des données se fait à l'importation des données. Contrairement aux fichiers texte qui acceptent un format libre, l'importation dans la base de données ne peut être faite que si la syntaxe textuelle est respectée. S'y ajoutent d'autres contraintes telles que l'existence du lemme comme forme, la conformité des étiquettes au référentiel. Celle-ci est aussi utilisée pour contrôler que la description de chaque forme contient au moins une étiquette indiquant sa catégorie grammaticale. Ces contraintes sont génériques alors que d'autres sont spécifiques au type de données importées, comme le fait de contrôler les nominalisations sur la présence d'un verbe et d'un nom d'après les informations déjà présentes dans la base. Cela implique évidemment de suivre un certain ordre de chargement des informations.

Les entrées lexicales qui ne correspondent pas strictement aux contraintes ne sont pas importées et sont marquées dans un log d'erreurs. Ces logs sont ensuite repris pour la correction des entrées incriminées.

Nous n'avons pas chargé les grammaires dans la base de données, mais prévu un mécanisme pour contrôler la validité des informations présentes dans les grammaires. La validation lexicale d'une grammaire revient à contrôler que toutes les informations lexicales des règles existent dans la base de connaissances lexicales. Le mécanisme est intégré à l'interface pour la validation lexicale des automates : après indication de l'ensemble de données à prendre en compte, l'interface liste toutes les informations lexicales présentes dans les règles et indique si elles ont été retrouvées dans la base de connaissances lexicales. Leur absence dans la base n'indique pas forcément une erreur, donc il faut vérifier la liste manuellement. Cette validation lexicale d'automates est particulièrement utile pour tracer des erreurs qui seraient noyées dans la masse d'informations autrement.

La validation lexicale du modèle de langage a été réalisée sur le même modèle sans avoir été intégrée à l'interface. Le modèle de langage est utilisé pour la désambiguïsation morphosyntaxique du lexique et contient un lexique et une grammaire. Les informations lexicales de la grammaire étant présentes dans le lexique, il suffit de soumettre ce dernier à une validation lexicale. Le lexique du modèle de langage se présente comme un ensemble de mots-formes avec leur catégorie grammaticale. Chaque combinaison est contrôlée en vérifiant la présence de ces éléments dans la base de connaissances lexicales.

En ce qui concerne les autres grammaires, notamment celles pour la désambiguïsation grammaticale à la requête, la question de la validation lexicale est de même importance, mais comme elles sont codées directement dans le code source, il est impossible d'automatiser la validation lexicale de la même façon.

7.2.5 Points forts du prototype

La qualité des ressources lexicales a été fortement améliorée grâce aux différents mécanismes de validation. Le chargement dans la base de données a indiqué un bon nombre d'entrées à corriger ou à regarder de plus près. A titre d'exemple, 694 entrées ont été signalées comme étant des doublons dans le lexique de base du français. Ces entrées n'étaient pas des lignes simplement dupliquées : l'ordre des étiquettes était différent, les rendant difficiles à détecter avec une commande de tri et de déduplication de type « sort -u ». Sur plus de 400 000 entrées, les entrées doublons étaient noyées dans la masse et n'auraient pu être décelées qu'avec des scripts dédiés et sous réserve d'avoir pensé à vérifier ce type de cas.

La validité lexicale des grammaires de détection d'entités et celle du lexique du modèle de langage sont devenues facilement et rapidement vérifiables, alors que l'écriture et l'exécution d'un script dédié étaient assez longues.

La facilité de la recherche multicritères renforce également la qualité des ressources, car on peut l'utiliser pour contrôler l'existence d'entrées qui mentionnent des combinaisons d'informations improbables ou interdites. L'une de ces combinaisons interdites est la présence des marques du singulier et du pluriel dans la même entrée. Les filtres permettent également

de sélectionner des listes d'entrées pour vérification manuelle, comme par exemple toutes les *singularia tantum*[103].

Les autres points forts par rapport au texte structuré sont les suivants.

- Pour la base de données, la structure fait partie du modèle, et le modèle fait partie de la structure. Les informations qui ne sont pas formatées comme le veut le modèle, ne peuvent être chargées dans la structure, contrairement aux fichiers texte, où tout est possible.
- Les informations sont uniques, c'est-à-dire qu'elles ne sont pas dupliquées entre plusieurs structures, ce qui évite de possibles divergences et augmente ainsi la qualité des données.
- Les lemmes ont des identifiants. En cas d'ambiguïté grammaticale, les lemmes obtiennent des identifiants différents, ce qui permet d'obtenir le paradigme de flexion homogène de chaque lemme.
- Une hiérarchie a été établie entre les catégories grammaticales et les traits morphosyntaxiques et sémantiques, alors qu'auparavant, il s'agissait d'une liste à plat. Les catégories et traits ne sont plus de simples suites de caractères, mais sont des éléments d'un ensemble fini connu.

La contrepartie de ces bénéfices est le prix de la mise en place et de l'entretien d'un logiciel, de la base de données et du serveur qui l'héberge. Cela représente un léger coût matériel et l'allocation d'un minimum de ressources humaines. Le retour sur investissement est néanmoins immédiat grâce à la qualité accrue des ressources. Les gains de temps réalisés au quotidien par les linguistes sont considérables. L'interface peut même être rendue disponible au service du support pourqu'ils deviennent plus autonomes au niveau de la qualification de problèmes liés aux ressources. Les coûts d'entretien du logiciel devraient être minimaux : une intervention ponctuelle devrait suffire pour adapter le chargement des données si leur format d'origine change.

7.2.6 Modification de l'architecture logicielle pour industrialisation

Maintenant que nous avons démontré que cet outil est indispensable pour une gestion efficace des ressources et que l'équipe linguistique a exprimé le besoin de pouvoir modifier les entrées affichées directement dans l'interface, Sinequa l'a industrialisé. Les fonctionnalités sont restées les mêmes, avec le projet d'ajouter la modification. Cette industrialisation est l'occasion de faire évoluer l'architecture de l'outil pour assurer son efficacité et garantir sa maintenabilité.

La nouvelle version de l'outil permet de charger en mémoire toutes les ressources lexicales mises à jour en local via SVN, de les consulter et de les mettre à jour via l'interface. Les ressources modifiées peuvent ensuite être retransmises au serveur SVN. Les ressources lexicales sont donc chargées en mémoire vive et non dans une base de données. Il y a quelques années, cette solution n'aurait pas été envisageable à cause d'une insuffisance en mémoire de travail des ordinateurs personnels.

Cette nouvelle architecture a l'avantage de ne pas modifier la chaîne de travail existante. Les fichiers texte ont des avantages indéniables. Leur extrême maniabilité pour des procédures automatiques par script, la grande facilité du suivi des modifications et d'archivage sont les

[103] Les *singularia tantum* sont les noms communs qui ne s'emploient qu'au pluriel (fiançailles, représailles, alentours, …) ou dont le sens du pluriel est différent du singulier (ciseaux, lunettes). Exemples pris de [Riegel et al., 1999], chap. VI, 3.4.

principales raisons pour lesquelles il est préférable de garder les ressources sous forme textuelle.

L'ajout de la fonction de modification dans la base de données pose également problème. A moins de complexifier la base avec des champs de gestion, ce qui alourdirait fortement son architecture et sa consultation, toute modification est répercutée directement dans la base. Ce fonctionnement n'est pas très adapté au mode de travail de l'équipe linguistique : les données sont souvent travaillées pendant de longues périodes et les modifications vérifiées plusieurs fois avant intégration au produit.

La base de données est sans aucun doute la meilleure solution dans des environnements multi-utilisateurs mis en place avec la constitution d'une ressource comme seule but. Le meilleur exemple est la constitution du lexique Celex, décrit dans [Wittenburg et al., 2004], où plusieurs utilisateurs travaillaient sur la même base de données centralisée. Le contexte à Sinequa n'est pas le même : peu de personnes travaillent sur les données, il est rare que les même personnes travaillent sur les mêmes données, et la mise à jour ne doit pas être immédiate dans la plupart des cas.

La nouvelle architecture réduit également les contraintes en termes de maintenance, d'hébergement et de sauvegarde qu'apporte une base de données. Initialement, il était question de remplacer les ressources sous forme de texte par la base de données et d'intégrer cette base directement au produit. Cela n'étant plus d'actualité, la base de données avait perdu tous ses avantages pour cet outil.

Quelques copies d'écran et une brève description de cet outil peuvent être trouvées en Annexe F, p. 261.

7.3 Conclusion

Dans ce chapitre nous avons passé en revue l'environnement que nous avons mis en place pour faciliter la mise à jour des ressources linguistiques. Il s'agit d'un ensemble d'outils nécessaires pour assurer cette opération de façon cohérente et efficace. L'outil qui donne un accès unique et globale à toutes les informations lexicales a été réalisé en utilisant le modèle de l'architecture linguistique. Il offre également une fonction pour vérifier si les informations lexicales présentes dans les grammaires de détection des entités nommées sont valables. Cet outil est actuellement en cours d'industrialisation. Cette réimplémentation doit assurer la maintenance à long terme du logiciel.

Certains outils existants peuvent bénéficier de la mise à disposition des informations lexicales par cet outil. L'éditeur d'automates pour les grammaires de détection d'entités nommées pourrait ainsi vérifier en temps réel les informations lexicales lors de l'édition des règles, ou proposer la fonctionnalité de vérification lexicale et générer un petit rapport sur demande.

La mise à jour serait bien facilitée si le lexique était géré en intension. Nous reviendrons sur ce sujet dans le chapitre sur l'acquisition, car les deux types d'opérations sont en effet bien liés à ce sujet.

Chapitre 8

Comprendre les ressources pour mieux les gérer

Un environnement de gestion de ressources ne facilite pas seulement la mise à jour. Il fournit également une vision sur l'état et l'évolution des ressources. Cette vision est nécessaire pour estimer le temps de travail que nécessite une tâche donnée. On peut ainsi s'inspirer du temps qu'une même opération a nécessité sur une tâche ou sur des données similaires. Dans ce chapitre nous présentons les outils que nous avons mis en place et qui procurent cette vision indispensable au responsable des ressources. Ils donnent un aperçu quantitatif global des ressources lexicales et procurent une vue qualitative sur les grammaires sous forme d'automates. Ils permettent également de suivre l'évolution des ressources lexicales.

8.1 Aperçu comparatif du nombre d'informations lexicales par langue

Afin d'obtenir une indication du niveau de développement des ressources lexicales de chaque langue, nous avons mis en place un aperçu qui donne le nombre de lemmes par catégorie pour chaque langue. L'idée sous-jacente est qu'à partir des indications fournies, le gestionnaire peut estimer la fiabilité et la complétude des lexiques.

Techniquement, nous avons intégré à l'arborescence de gestion des ressources un ensemble de scripts PERL qui font le calcul à partir des fichiers texte et créent des fichiers XML. Ceux-ci sont affichés dans un navigateur web à l'aide d'une feuille de style XSL.

La figure 59 montre une vue partielle de cet aperçu : la première colonne indique le type d'information calculé, la deuxième des estimations de minima et la troisième les chiffres du lexique général néerlandais. L'aperçu montre normalement les informations pour les 13 langues européennes que nous gérons[104]. En raison de la confidentialité de ces données stratégiques, nous nous limitons ici au néerlandais.

	min	NL
lemmes sans PPS	50000	**81008**
lemmes -PPS -OUI -MC	40000	**50120**
(lemmes)CAT		81008
%formes		217595
descriptions		265373
NOM	30000	**60072**
ADJ	10000	**9326**
VER	5000	**9371**
ADV	1000	**1413**
PRO		159
PPS		0
PRE		114
DET		54
CSU		56
INTERJECTION		124
CCO		18
PV		111
QUA		91
N_A		0
MC		30870
lemmes sans MC		50138
OUI,NON,OUIN		18
lemmes sans OUI		80990

Figure 59 : Aperçu comparatif du nombre d'informations lexicales par langue (NL)

Les deux premières lignes du tableau montrent le nombre de lemmes avec ou sans mots composés, en l'occurrence respectivement 81 008 et 50 120. Les trois lignes suivantes indiquent le nombre de lemmes, de mot-formes et de descriptions. A partir de ces informations, il est possible de calculer le ratio mot-formes/lemme par langue. Ensuite, par

[104] DA, DE, EL, EN, ES, FI, FR, IT, NL, PL, PT, RU, SV

catégorie est donné le nombre de lemmes, en commençant par les quatre catégories les plus nombreuses : nom (NOM), adjectif (ADJ), verbe (VER) et adverbe (ADV). Suivent ensuite les catégories grammaticales des mots grammaticaux : pronom (PRO), préposition (PRE), etc. Les étiquettes MC, OUI, NON et OUIN sont des indicateurs de mots composés, compositionnels pour MC, et avec espaces ou tiret pour les autres.

Nous faisons donc plusieurs distinctions pour le calcul du nombre de lemmes : nom propre ou non, mot simple ou mot composé, et puis par catégorie. Etant construit à partir du lexique de base, le tableau ne contient que des mots communs. Si on ajoutait les noms propres ; les chiffres sur les noms deviendraient plus difficiles à interpréter et à comparer, d'autant plus que les lexiques de noms propres sont inégalement développés dans les différentes langues. La distinction entre mot composé et mot simple est importante dans le sens où la comparaison de leurs chiffres ne donne pas les mêmes informations. Les mots composés étant composés de mots simples, il est important de pouvoir les compter indépendamment.

La colonne « min » dans la figure 59 est une colonne de minima : elle contient des nombres que nous estimons être le nombre minimum de lemmes requis par catégorie pour notre type d'application. Ces chiffres ne sont pas le résultat de savants calculs, mais d'une estimation subjective à partir des chiffres de toutes les langues, en donnant plus ou moins de poids aux langues selon les développements faits pour chaque langue.

D'après ces estimations, le lexique de base de chaque langue devrait contenir au moins 50 000 lemmes au total, dont 40 000 sont des mots simples. Il contient au moins 30 000 noms, 10 000 adjectifs, 5 000 verbes et 1 000 adverbes. Pour le nombre de lemmes, de noms, d'adjectifs et de verbes, ces chiffres sont environ un tiers en dessous des moyennes de toutes les langues. Pour les adverbes, nous avons fixé le seuil minimal à seulement un quart de la moyenne en raison des très fortes disparités constatées entre les langues. Nous en avons conclu que le nombre d'adverbes est plus dépendant des caractéristiques de la langue que pour les autres catégories.

Au début, nous avions mis les moyennes au lieu des minima, mais cela ne correspondait pas aux objectifs de gestion. En effet, si on augmente le nombre de noms d'une langue déjà bien fournie, cela fait monter le nombre souhaité de noms dans toutes les langues, et d'autres langues bien fournies peuvent ainsi être mises en défaut. Le but est seulement d'indiquer des possibles lacunes. Nous avions aussi pensé à mettre des proportions, mais comme certaines classes sont plus « ouvertes » que d'autres, c'est-à-dire qu'il y a plus de nouveaux mots qui apparaissent dans une catégorie particulière que dans les autres, cela risque de disproportionner le schéma.

Dans un monde idéal, nous disposerions des chiffres indiquant la proportion idéale de chaque catégorie syntaxique par rapport au nombre de lemmes total, et ils varieraient en fonction de la taille du lexique et de la langue. Ce genre de tableau n'existe pas, et il est impossible de le dresser sans introduire de biais. Ce biais serait lié au calcul de la couverture, qui est forcément lié à un corpus, et aucun corpus ne peut prétendre à représenter la langue. Même si on peut dresser ce tableau pour un corpus donné, ce corpus ne serait sans doute pas représentatif pour les corpus que nous indexons. Nous n'avons pas de moyen de vérifier la représentativité, car nous n'avons généralement pas accès aux corpus des clients.

Même s'il faut l'interpréter avec précautions, notre aperçu comparatif donne une indication grossière mais précieuse sur l'état des ressources lexicales. Pour que la comparaison entre les ressources prenne pleinement sens, il faut prendre en considération les remarques suivantes :

- Les chiffres sont plus comparables entre langues similaires[105] : comme elles mettent en œuvre les mêmes mécanismes linguistiques au niveau lexical ou grammatical, les choix d'arbitrage entre le lexique et la grammaire sont en général les mêmes. Au niveau du nombre de mots dans le lexique, on peut donc supposer que les nombres absolus et relatifs entre ces langues devraient être semblables. De gros écarts peuvent indiquer des manques substantiels.
- Les chiffres absolus ne disent rien sur la couverture du lexique. La couverture est calculée par rapport à un corpus, qui est alors pris comme référence. Or, il est impossible de construire un corpus de référence qui reflète les caractéristiques des corpus clients, puisque nous n'y avons pas accès. Il est néanmoins possible d'avoir des lignes directrices. Il semble par exemple évident qu'il est plus important de couvrir les mots fréquents d'une langue plutôt que les mots rares. Les mots grammaticaux sont alors très importants, si nous partons du principe que ce sont pour la plupart des mots fréquents[106] et qu'ils sont tous couverts par le lexique, nous pouvons comparer leurs nombres absolus. Cela ne nous permet pas vraiment d'arriver à des conclusions nettes, sauf encore une fois si les chiffres montrent de très grands écarts entre langues similaires et que les mécanismes des langues comparées sont similaires.

En aucun cas il ne convient d'utiliser ces tableaux pour en déduire des conclusions sur le nombre de mot-formes ou de proportions entre catégories d'une langue dans l'absolu. Nos lexiques ne servent pas à la description de la langue mais forment un équilibre avec les règles pour les traitements passés en revue dans la partie 3.

8.2 Calcul de la complexité des grammaires

Il est important d'obtenir une vision de la complexité des automates pour être en mesure de planifier des travaux et d'attribuer des ressources humaines en tenant compte de la complexité des données concernées. Le nombre d'automates ne dit rien sur la complexité de ceux-ci, alors que des automates simples sont évidemment plus faciles à gérer que des automates complexes.

Dans un souci de bonne compréhension, la terminologie que nous utilisons par la suite est la suivante : un automate est fait d'au moins trois nœuds, dont un nœud initial et un nœud terminal, les autres nœuds exprimant les conditions de transition d'un nœud à un autre. Des flèches, appelées transitions, relient les nœuds entre eux et indiquent les transitions possibles.

Nous n'avons pas trouvé de référence de travaux étudiant la complexité des automates du point de vue de leur gestion. Un sujet similaire est néanmoins discuté en théorie des jeux où l'on implémente parfois des stratégies de jeu par des automates. La complexité de la stratégie étant d'importance, [Kalai et Stanford, 1988][107] a introduit la mesure du nombre d'états pour indiquer la complexité de l'automate minimal implémentant la stratégie. Dans le souci d'améliorer cette mesure, [Banks et Sundaram, 1990] propose de tenir également compte de la complexité transitionnelle de l'automate, c'est-à-dire du nombre de transitions. Pour éviter un calcul complexe, les auteurs posent qu'un automate est plus complexe qu'un autre s'il est au

[105] Nous préférons l'expression *langues similaires* à *langues de même famille*, car deux langues de la même famille peuvent fonctionner assez différemment alors que deux langues de familles différentes peuvent présenter de fortes similitudes dans leur fonctionnement.
[106] Les mots grammaticaux ne sont pas un strict sous-ensemble des mots les plus fréquents.
[107] Ces références semblent datées, mais les méthodes sont toujours utilisées dans des articles récents comme par exemple [Lee et Sabourian, 2007].

moins également ou plus complexe en ce qui concerne la complexité transitionnelle ou le nombre d'états, et strictement plus complexe sur l'autre critère. Comme les auteurs de [Banks et Sundaram, 1990] l'admettent eux-mêmes, leur mesure est très imparfaite, notamment du fait qu'elle n'admet pas qu'un automate avec peu d'états mais beaucoup de transitions soit considéré comme plus complexe qu'un automate avec beaucoup d'états mais peu de transitions.

Ces mesures étant rudimentaires et insuffisantes pour estimer la complexité des automates du point de vue de leur gestion, nous avons développé une nouvelle mesure de complexité d'automate. Le nombre de nœuds et de transitions ne sont pas de bons indicateurs de la complexité d'un automate : ils n'indiquent en rien le nombre de règles. Un ensemble de nœuds et de transitions qui forment un automate peut cacher entre 0 et un nombre infini de règles si on admet des cycles. Mieux vaut donc comptabiliser le nombre de chemins, un chemin étant égal à une règle, et, compter les cycles (qui sont fortement déconseillés dans une application industrielle) un nombre limité de fois.

Ce propos est illustré dans le tableau 10, où nous affichons le nombre de nœuds, de transitions et de chemins pour les automates des groupes nominaux courts dans les cinq principales langues européennes (février 2009). S'il y en a, les boucles et cycles sont pris en compte deux fois. On y voit notamment que le nombre de chemins n'est pas fonction du nombre de nœuds et de transitions : les nombres de nœuds et des transitions sont comparables en anglais et en espagnol, mais le nombre de chemins est totalement différent. Si certains automates que nous gérons contiennent seulement quelques chemins, d'autres en comptent des milliers.

	Nœuds	Transitions	Chemins
DE	22	35	17
EN	61	112	884
ES	58	102	175
FR	154	379	1424
IT	74	127	285

Tableau 10 : Quantification d'un automate

Le nombre de chemins indique la taille de l'automate, mais pas pour autant sa complexité du point de vue de la gestion. Pour estimer cette complexité, nous avons mis au point une formule, basée sur les observations suivantes :

1. Un automate est simple, donc plus facile à gérer, si le nombre de chemins est proche ou inférieur au nombre de transitions (comme c'est le cas pour l'allemand dans le tableau 10). Cela indique une linéarité des chemins. La complexité augmente avec l'écart entre les deux. On obtient cette estimation de la complexité en divisant le nombre de chemins par le nombre de transitions. Plus elle est élevée, plus l'automate est complexe.

2. Un automate est simple si le nombre de transitions est proche du nombre de nœuds. La complexité s'accroît avec un nombre plus élevé de transitions si le nombre de nœuds reste constant. On obtient cette valeur en divisant le nombre de transitions par le nombre de nœuds.

L'addition n'étant pas assez discriminante pour des grands nombres nous combinons ces deux valeurs par multiplication pour obtenir une valeur de complexité d'automate. Après simplification, la complexité correspond au ratio entre le nombre de chemins et le nombre de nœuds. Nous avons ajouté la racine carrée pour réduire l'étendu des chiffres obtenus. Les automates les plus simples sont donc ceux dont la complexité s'approche le plus de 1.

$$Complexité(Aut) = \sqrt{\frac{\#chemins(Aut)}{\#transitions(Aut)} \times \frac{\#transitions(Aut)}{\#noeuds(Aut)}}$$
$$= \sqrt{\frac{\#chemins(Aut)}{\#noeuds(Aut)}}$$

Les valeurs ainsi obtenues sont données dans la figure 60, regroupées selon le type d'automates. On peut globalement voir des différences par types d'automates. Les automates de noms géographiques sont ainsi les plus « simples » à gérer, ensuite les groupes nominaux courts, suivis par ceux des entreprises. Les automates de détection des personnes et des groupes nominaux longs terminent le palmarès et correspondent à des automates complexes à gérer. Ces derniers tiennent cette place à cause de la présence de boucles, ce qui augmente considérablement le nombre de chemins, même si elles ne sont comptabilisées que deux fois. En effet, ce sont les seuls automates où nous avons *autorisé* la présence de boucles.

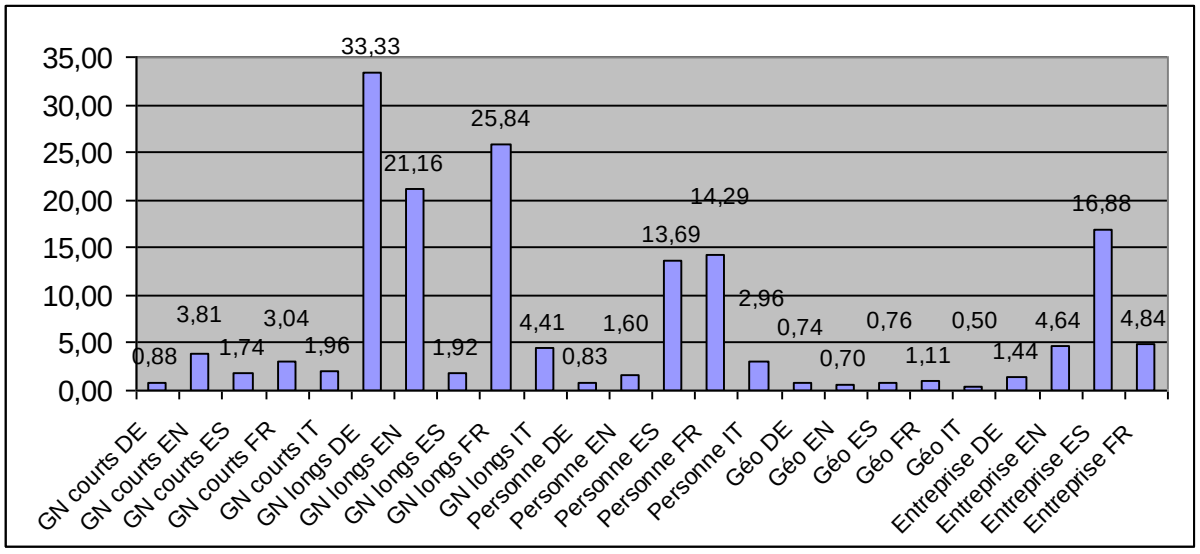


Figure 60 : Complexité des automates du point de vue de la gestion, regroupé par type

D'après ce que nous pouvons voir dans les illustrations de la figure 61, le calcul de la complexité correspond aux rendus visuels des automates. Les captures d'écran miniatures des 5 automates des groupes nominaux courts, dont nous complétons les chiffres dans le tableau 11 avec la complexité, rendent bien compte des différences calculées. On voit que l'allemand (DE) est bien l'automate le plus simple du point de vue du rendu visuel, suivi par l'espagnol (ES) et l'italien (IT). Le français a l'air très complexe visuellement (éclaté sur deux copies d'écrans : FR1 et FR2), ce qui se vérifie dans les chiffres. Le calcul de complexité peut cependant modifier l'interprétation d'un automate dont le rendu visuel peut paraître relativement simple. C'est le cas de l'anglais (EN), dont le rendu visuel semble innocent, mais le chiffre de la complexité montre que l'automate n'est pas du même calibre que l'espagnol et l'italien : en effet, les chemins sont beaucoup moins linéaires. Cela veut dire que s'il faut retravailler cet automate, il faut prévoir plus de temps que pour l'allemand, l'espagnol ou l'italien. Pour la gestion, cette information est d'une grande utilité.

	Nœuds	Transitions	Chemins	Complexité
DE	22	35	17	0,88
EN	61	112	884	3,81
ES	58	102	175	1,74
FR	154	379	1424	3,04
IT	74	127	285	1,96

Tableau 11 : Complexité des automates des groupes nominaux courts

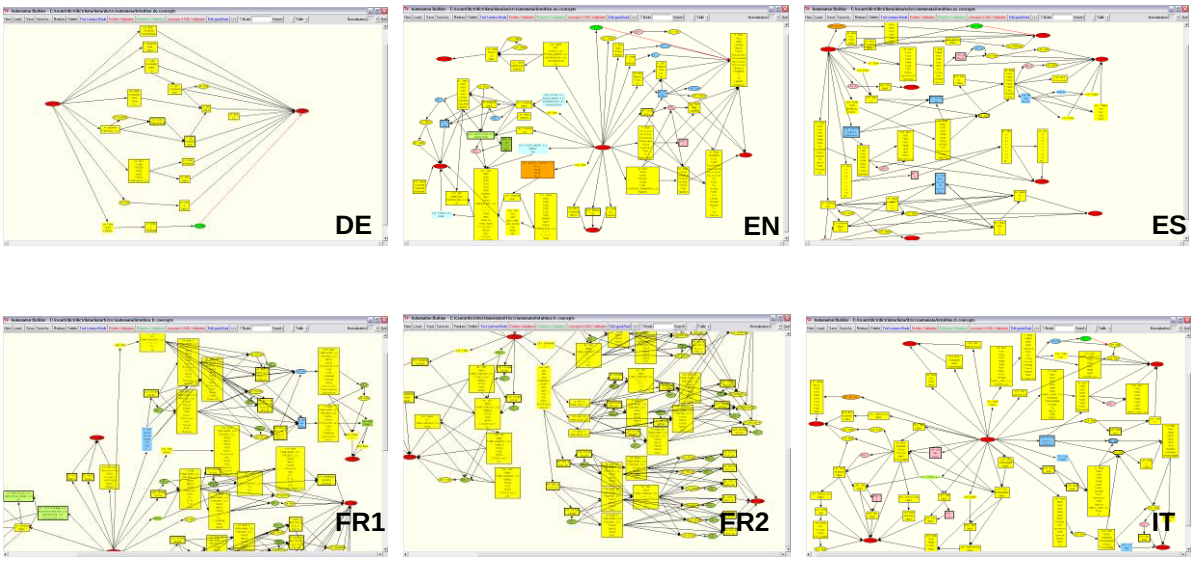


Figure 61 : Captures d'écran des automates des groupes nominaux courts

Comme nous l'avons déjà expliqué, un chemin correspond à une règle. Dans les suites de validation que nous avons mises en place, il existe au moins un exemple pour chaque règle. Elles contiennent donc au minimum autant d'exemples que de règles, car on ne supprime jamais d'exemples dans la suite de validation pour garantir la non régression de la nouvelle version de l'automate.

Cette mesure nous donne une vision plus objective sur la complexité des automates et en conséquence nous permet d'estimer le temps nécessaire pour prendre en main ces automates. Cette information est essentielle pour une bonne planification.

8.3 Un regard sur l'évolution des ressources

Les ressources évoluent beaucoup au cours de la mise en place des traitements, mais aussi quand leurs spécifications sont redéfinies. Certaines ressources sont par nature évolutives, comme par exemple les lexiques de grande couverture qui sont par définition incomplets ou les grammaires d'extraction d'entités dont la qualité peut être continuellement améliorée. Cette mise à jour est faite, au fil des années, par beaucoup de personnes différentes, ce qui nécessite un encadrement strict. Dans ce chapitre nous illustrons ce point en montrant combien de personnes ont travaillé sur les ressources de Sinequa sur une période d'un peu plus de trois ans. Ensuite, en absence d'un système de gestion qui trace de façon détaillée toutes les modifications faites, nous proposons une méthode pour quantifier l'évolution des lexiques à partir des informations fournies par le système de gestion des versions.

8.3.1 Un grand nombre d'auteurs

Pendant la durée de notre thèse, nous avons encadré un grand nombre de personnes, dont des salariés et des stagiaires. Le système de *versioning*[108] que nous avons mis en place pour la gestion des lexiques et grammaires, nous permet d'indiquer le nombre de *commits* par date et par auteur (voir figure 62). Un *commit* est l'action d'envoyer des données locales au serveur de données. Les données sont ainsi centralisées et toute personne qui les utilise peut les mettre à jour en passant par le serveur.

Il ne faut pas espérer déduire des informations fiables sur l'évolution des ressources de ces tableaux, car un seul commit peut cacher un grand comme un petit nombre de modifications. Le graphique donne néanmoins une bonne idée du grand nombre de commits et l'implication des différentes personnes dans le processus d'acquisition et de mise à jour sur la période qui nous est accessible pour cette étude[109]. Pour certains salariés et pour les stagiaires, on peut en déduire le trimestre d'arrivée et/ou de départ. Ordonnés par nombre croissant des modifications faites au total par personne (voir figure 63), les six premiers contributeurs sont sans surprise des salariés de Sinequa. Les autres contributeurs sont des stagiaires à deux exceptions près.

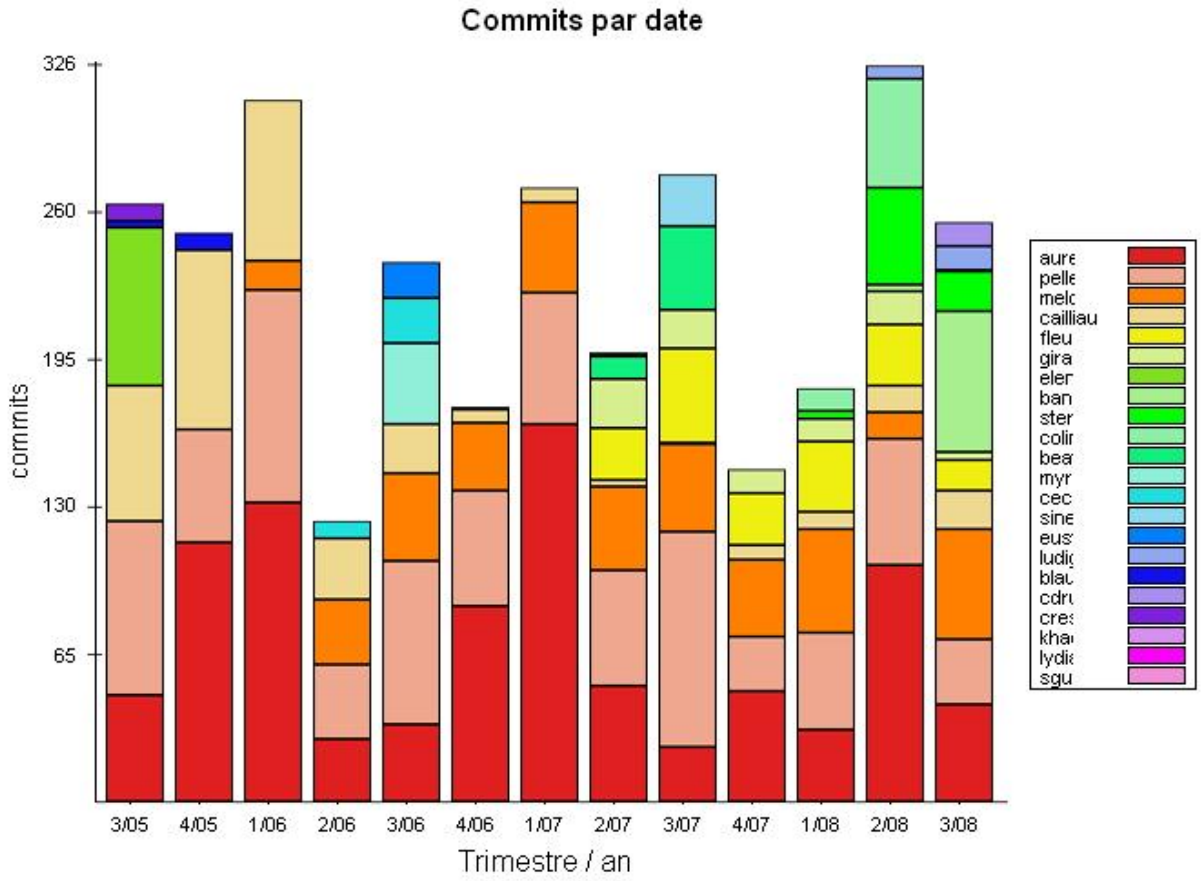


Figure 62 : Nombre de commits sur SVN par personne par trimestre sur la période 7/2005 - 10/2008

[108] Techniquement il s'agit de CVS, remplacé en 2005 par SVN.
[109] Depuis la migration de CVS vers SVN jusqu'au changement des logins pour des raisons techniques.

Le système de versioning nous procure encore les informations du tableau 12 pour la même période.

Nombre de semaines	170
Nombre d'auteurs	22
Nombre de commits	3031
Nombre de changements de fichier	8105

Tableau 12 : Informations procurées par le système SVN

Sur trois ans et quelques mois, pas moins de vingt-deux personnes ont donc travaillé sur les mêmes ressources. Les 6 salariés les plus actifs en termes de commits en ont effectué 85 %.

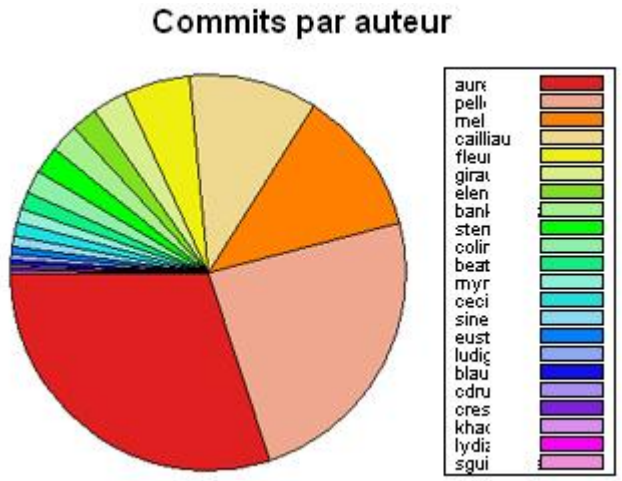


Figure 63 : Proportion des commits sur SVN par personne cumulés sur la période 7/2005 - 10/2008

Les informations que nous procure le système de suivi des versions ne nous suffisent clairement pas. Nous voulons savoir quelles ressources ont été modifiées pour mieux comprendre la vie des ressources.

Deux types de ressources que nous avons manipulées se prêtent à une quantification de l'évolution car ils sont plus sujets que les autres à des mises à jour : les grammaires sous forme d'automate et les lexiques morphosyntaxiques. Nous nous concentrerons sur les lexiques, car les automates ont évolué trop massivement durant les années pour pouvoir quantifier leur évolution. Deux raisons expliquent le grand nombre de modifications qui ont eu lieu sur les automates : la mise au point de la stratégie qui est passée par plusieurs approches déployées successivement, et d'autre part l'évolution des spécifications qui ont suivi l'évolution du produit.

8.3.2 Suivi de l'évolution des lexiques morphosyntaxiques

Un lexique évolue au cours de sa vie. Ses entrées sont l'objet de trois modifications : l'ajout, la suppression et la correction. Cette dernière est en réalité une combinaison particulière de suppression et d'ajout : les entrées supprimées et ajoutées ont une partie de l'entrée en commun.

Le suivi de l'évolution des lexiques est important car il nous donne, au-delà d'une vue globale sur l'évolution des lexiques en soi déjà intéressante, une idée du travail fait par les linguistes manipulant les lexiques.

8.3.2.1 Méthode de calcul de l'évolution d'un lexique

Pour suivre l'évolution d'un lexique, il faut comparer les versions successives et identifier les modifications. La différence du nombre d'entrées ne donne qu'une vue très partielle sur les modifications, car elle n'indique ni les corrections ni les suppressions, et pour cela le chiffre des ajouts n'est pas fiable. Une comparaison de fichiers avec la commande Unix *diff* donne des informations sur les modifications, mais pas sur la nature de la correction. En outre, il demande un prétraitement considérable sur la syntaxe des fichiers. En effet, la comparaison ligne par ligne demande que les entrées soient dans le même ordre et que la syntaxe des fichiers soit identique. Ce n'est pas le cas de nos lexiques, où plusieurs syntaxes s'équivalent et dans lequel l'ordre des entrées n'a aucune importance.

Pour ces raisons, nous avons mis en œuvre un calcul spécifique. Nous l'avons appliqué sur des lexiques morphosyntaxiques dont les entrées sont constituées d'un mot-forme, d'un lemme et d'une description, mais notre méthode est généralisable à des lexiques contenant d'autres informations.

Pour calculer les différences entre les versions successives d'un lexique, on liste tout d'abord les informations à prendre en compte. Dans notre cas il s'agit du mot-forme, du lemme et de la description. Il serait possible de décomposer la description, mais nous ne voulons pas aller aussi loin dans le détail. Ensuite, en énumérant toutes les combinaisons de modifications d'une entrée, il faut analyser chaque combinaison et en déduire sous quelles conditions il s'agit d'un ajout ou d'une correction.

Nous donnons cette analyse pour un lexique morphosyntaxique dans le tableau 13, dans lequel Lexique2 est une version postérieure de Lexique1. Les suppressions sont calculées à la fin, en soustrayant le nombre d'entrées identiques et de corrections du total d'entrées du Lexique1. Cela correspond également à la différence entre le nombre d'ajouts et la différence en nombre d'entrées des Lexique1 et du Lexique2. A partir de cette analyse on déduit l'algorithme à utiliser pour le calcul de l'évolution.

Mot-forme	Lemme	Description	Type de modification	Analyse
=	=	=	0	Entrées identiques
=	=	≠	1	Ajout si le nombre de combinaisons de mot-forme + lemme de Lexique1 est inférieur à celui de Lexique2.
				Correction si le nombre de combinaisons de mot-forme + lemme de Lexique1 est supérieur ou égal à celui de Lexique2.
=	≠	=	2	Ajout si le nombre de combinaisons de mot-forme + lemme de Lexique1 est inférieur à celui de Lexique2.
				Correction si le nombre de combinaisons mot-forme + lemme de Lexique1 est supérieur ou égal à celui de Lexique2.
≠	=	=	3	Ajout si le nombre de combinaisons de lemme + description de Lexique1 est plus petit que celui de Lexique2.
				Correction si le nombre de combinaisons lemme + description de Lexique1 est supérieur ou égal à celui de Lexique2.
=	≠	≠	4	Ajout si le nombre de mot-formes de Lexique1 est inférieur à celui de Lexique2.
				Correction si le nombre de mot-formes de Lexique1 est supérieur ou égal à celui de Lexique2.
≠	=	≠	5	Ajout si le nombre de lemmes de Lexique1 est inférieur à celui de Lexique2
				Correction si le nombre de lemmes de Lexique1 est supérieur ou égal à celui de Lexique2
≠	≠	=	6	Ajout
≠	≠	≠	7	Ajout

Tableau 13 : Analyse des différentes combinaisons de modification d'un lexique morphosyntaxique

Dans tous les cas, la différence entre ajout et correction est approximative, car il est impossible d'identifier exactement l'opération qui a été effectuée. Même si nous avons exploité toutes les informations à notre disposition pour essayer de faire la différence, notre méthode n'est pas exacte. Par exemple, si la forme et le lemme sont différents et la description la même, nous avons estimé qu'il s'agit d'un ajout. Néanmoins, il y aura des cas pour lesquels il s'agit d'une correction, mot-forme et lemme nécessitant étant corrigés

conjointement comme on peut le voir dans le tableau 14 où *F* est le mot-forme, *L* le lemme et *D* la description d'une même entrée du lexique.

Lexique1 (version 3524)	Lexique2 (version 3599)
F: Stodiek Eurpoa Immobilien *L*: Stodiek_Eurpoa_Immobilien *D*: NOM.NPR.OUI.RCAP.ENTREPRISE	*F*: Stodiek Eur**op**a Immobilien *L*: Stodiek_Eur**op**a_Immobilien *D*: NOM.NPR.OUI.RCAP.ENTREPRISE
F: TC Unterhal Tungselektronik *L*: TC_Unterhal_Tungselektronik *D*: NOM.NPR.OUI.RCAP.ENTREPRISE	*F*: TC Unterha**lt**ungselektronik *L*: TC_Unterha**lt**ungselektronik *D*: NOM.NPR.OUI.RCAP.ENTREPRISE
F: Valora Effekten *L*: Valora_Effekten *D*: NOM.NPR.OUI.RCAP.ENTREPRISE	*F*: Valora Effekten **Handel** *L*: Valora_Effekten**_Handel** *D*: NOM.NPR.OUI.RCAP.ENTREPRISE

Tableau 14 : Exemple de modification conjointe de mot-forme et de lemme entre deux versions du même lexique

Un calcul de similarité avec une distance de type Levenshtein pourrait s'appliquer pour discriminer les corrections des ajouts/suppressions. Nous ne sommes pas allé aussi loin dans ce cas précis, car leur nombre est marginal comme nous avons pu constater en étudiant les lexiques. Quand ces cas se présentent, nous les comptons donc comme autant de suppressions et d'ajouts.

8.3.2.2 A chaque lexique son histoire

A titre d'illustration, nous avons calculé l'évolution des lexiques de grande couverture de Sinequa pour les cinq langues suivantes : allemand, anglais, espagnol, français et italien, sur les 7 semestres de la période de fin août 2005 à fin mars 2009. Le découpage temporel est illustré dans le tableau suivant.

31/08/05 - 31/03/06	31/03/06 - 2/10/06	2/10/06 - 2/04/07	2/04/07 - 1/10/07	1/10/07 - 1/04/08	1/04/08 - 2/10/08	2/10/08 - 31/03/09
S1	S2	S3	S4	S5	S6	S7

Tableau 15 : Les périodes et leurs identifiants de semestre correspondants

Dans la suite l'évolution des lexiques de chaque langue est visualisée et mise en contexte. La couverture des semestres ne coïncide pas avec l'année civile, mais avec les périodes susceptibles ou non de montrer une activité supérieure due à la présence de stagiaires.

a. Le lexique français

L'évolution du lexique français semble assez soutenue, mais on identifie tout de même deux longues périodes d'activité séparées par un semestre de basse activité. La première période

(S1-2) est marquée par un très grand nombre de suppressions accompagné et suivi par des corrections et des ajouts. La deuxième période (S3-6) marque une évolution modeste.

	S1	S2	S3	S4	S5	S6	S7
Ajouts	4206	16006	1077	1535	3058	6535	78
Corrections	10867	6137	885	4695	2530	4821	63
Suppressions	45434	5343	1019	3457	4714	2670	17
Total	60507	27486	2981	9687	10302	14026	158

Tableau 16 : Evolution du lexique morphosyntaxique français (septembre 2005 – mars 2009)

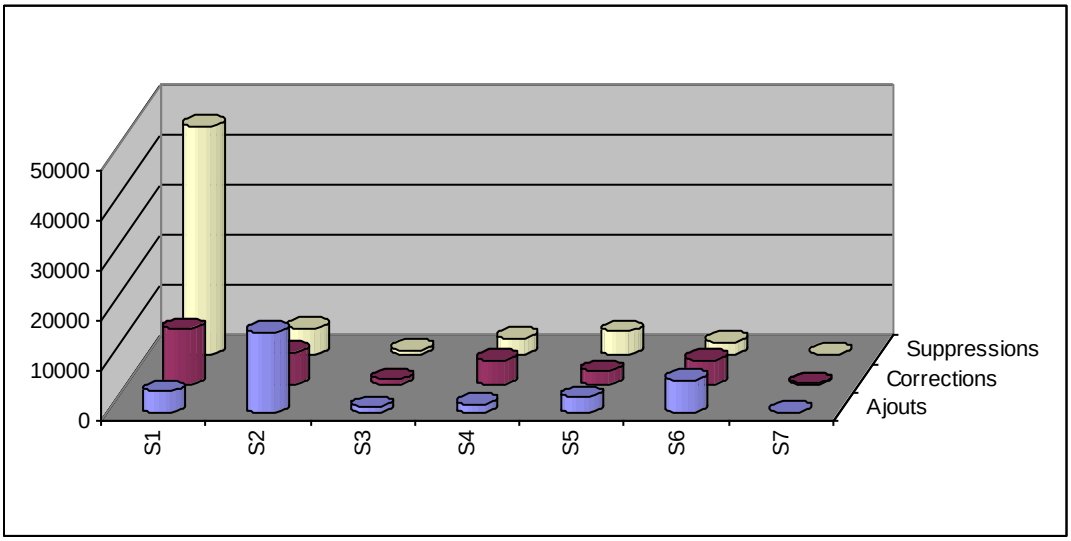


Figure 64 : Evolution du lexique morphosyntaxique français (septembre 2005 – mars 2009)

Le grand nombre de suppressions du premier semestre (S1) est en premier lieu dû à la mise à jour de la liste des noms de villes français et internationaux. Un changement dans les spécifications de l'extraction des noms de villes a mené à la suppression de bon nombre d'entre eux suivant un tri selon le nombre d'habitants. Les mots capitalisés ont également fait l'objet d'un nettoyage. Un certain nombre de noms propres, comme *Balladur*, *Bangemann*, *Bashung*, *Crédit Lyonnais*, *Polanski*, *Pollack*, *Pompidou*, étaient présents dans les lexiques. Or ces noms étaient hérités d'un usage révolu dans un correcteur orthographique, d'où le marquage indiquant de capitaliser le mot s'il ne l'était pas. Enfin, suite au constat que les abréviations sont presque toujours trop ambiguës pour pouvoir servir dans toutes les configurations, la plupart des abréviations ont également été évacuées du lexique.

Une des corrections importantes est le changement du codage des noms propres géographiques pour ajouter le marquage obligeant à tenir compte de la capitalisation à l'étiquetage morphosyntaxique. Dans le même style, la reconnaissance des entreprises a été relâchée, le marquage passant d'un respect strict de la casse à la capitalisation uniquement lors de l'étiquetage.

Dans la même période, l'organisation du codage sémantique sur certaines entrées principalement géographiques a changé. Ces codes étant utilisés dans la détection des entités géographiques, ils ont été remaniés pour correspondre aux nouvelles spécifications. *Afrique du Nord* par exemple était codé comme un *pays*, alors que sa description change en *région*. Un nouveau codage comme *département* apparaît alors que ces mots étaient codés comme

région auparavant. Encore dans la sémantique, on remarque l'ajout assez massif d'un codage comme *lieu* d'endroits aussi variés qu'*avion, bulldozer, baleinière* ou encore *bananier*, ce qui fait supposer qu'il s'agit d'une fusion d'informations préexistantes et non pas d'un codage à la main.

Dans les ajouts, on compte des noms de journaux et les noms officiels des pays, qui sont souvent des expressions, comme par exemple la *République somalienne*.

Le deuxième semestre de cette première période (S2) est marqué par une mise à jour des noms d'entreprises, entraînant un ajout massif d'entreprises ainsi qu'un nombre élevé de suppressions. Par ailleurs, la description des noms d'entreprises a été modifiée par l'ajout du marquage obligeant à tenir compte de la capitalisation des noms d'entreprises lors de l'étiquetage. Tous les noms d'entreprise ont donc été l'objet d'une modification.

A part cela on note la trace de plusieurs petits chantiers. Dans la continuité du précédent semestre, il y a quelques changements dans le marquage sémantique concernant l'étiquette *habitant* et sa différenciation avec la *nationalité*. La description des pays subit encore des corrections, par exemple *Angleterre* qui était un *pays* devient une *région* (du Royaume-Uni). Toujours dans les corrections, on trouve un travail très linguistique sur les « noms adjectifs »[110], dont certains faisaient à tort partie de cette classe (par exemple *sanguin*). Pour certains mots, la relation entre masculin et féminin du même nom était traité au niveau du lemme, alors que ce n'est pas le bon niveau pour une application de moteur de recherche. On l'a donc transféré au niveau des renvois sémantiques. Par conséquent, le lemme des mots désignant des personnes a été changé pour les formes féminines : *installateur* et *installatrice* n'ont par exemple plus le même lemme *installateur*, mais deux lemmes différents *installateur* et *installatrice*. Certains noms de pays sont remodifiés (par exemple *République somalienne* devient *République démocratique somalienne*). Et finalement on note la suite de la suppression d'abréviations jugées trop ambiguës.

Les trois semestres de la deuxième période (S4-S6) ne sont pas vraiment marqués par un événement en particulier, mais plutôt par des modifications en continu. Elles concernent presque exclusivement la détection des différentes entités nommées. Le premier semestre (S4) est marqué par la suppression des prénoms composés dont la détection est passée au niveau des règles de la grammaire. Des prénoms sont ajoutés et anciennes et nouvelles entrées de prénoms obtiennent un marquage imposant le respect de la capitalisation à l'étiquetage morphosyntaxique. Finalement, les catégories grammaticales des composants des mots composés sont spécifiées pour ajouter une désambiguïsation des composants si c'est nécessaire, par exemple nom et adjectif pour *acquis social*, le nom pouvant également être interprété comme un participe de verbe.

Au deuxième semestre (S5) un marquage d'ambiguïté a été ajouté sur les villes. Le lien entre le nom long et le nom court du pays (par exemple *République démocratique du Timor Oriental* et *Timor oriental*) était fait à travers le lemme, ce qui a été supprimé. Ce lien a été réintroduit en même temps au niveau des renvois sémantiques, qui ne font pas partie de ces lexiques-ci. Les noms de journaux ont été supprimés suite à la décision de ne plus les reconnaître dans les entreprises. Certains noms de profession faisant double emploi avec des entrées déjà existantes ont également été supprimés, ce marquage sémantique n'étant plus utilisé pour la reconnaissance de personnes.

---

[110] Il existe une classe spéciale pour les noms qui peuvent être utilisés comme adjectifs et vice versa, comme *douteur-douteuse, dreyfusard-dreyfusarde, droitier-droitière, vrai*, etc. Techniquement parlant, il s'agit d'une recatégorisation grammaticale, ce qu'on appelle la *conversion* ([Riegel et al., 1999], chap. XVII, 3.4).

Le dernier semestre (S6) de cette période est marqué par un travail exclusif sur les noms d'entreprises, avec un nombre d'ajouts relativement important. Le marquage concernant le respect de la casse à l'étiquetage morphosyntaxique est rationalisé en remplaçant un double marquage par un simple. Le marquage du singulier est supprimé sur les quelques entreprises qui le portaient pour renforcer la cohérence du codage lexical.

b. Le lexique anglais

L'évolution du lexique morphosyntaxique anglais est indiquée dans le tableau 17 et illustrée dans la figure 65. Elle est globalement constante, sauf pour deux phases. La première (S1-S2) est caractérisée par un nombre très important de suppressions et un plus grand nombre de modifications sur la même période, suivi dans la période suivante d'un grand nombre d'ajouts et d'un nombre toujours élevé de suppressions. La 2e phase (S5) est caractérisée par un grand nombre de corrections seulement. Remarquons la constance des ajouts du S3 au S7.

	S1	S2	S3	S4	S5	S6	S7
Ajouts	4003	25 864	2451	2840	2615	2694	2163
Corrections	4537	1274	871	1204	9416	2426	43
Suppressions	47 794	3760	2011	1590	1885	1038	87
Total	56 334	30 898	5333	5634	13 916	6158	2293

Tableau 17 : Evolution du lexique morphosyntaxique anglais (septembre 2005 – mars 2009)

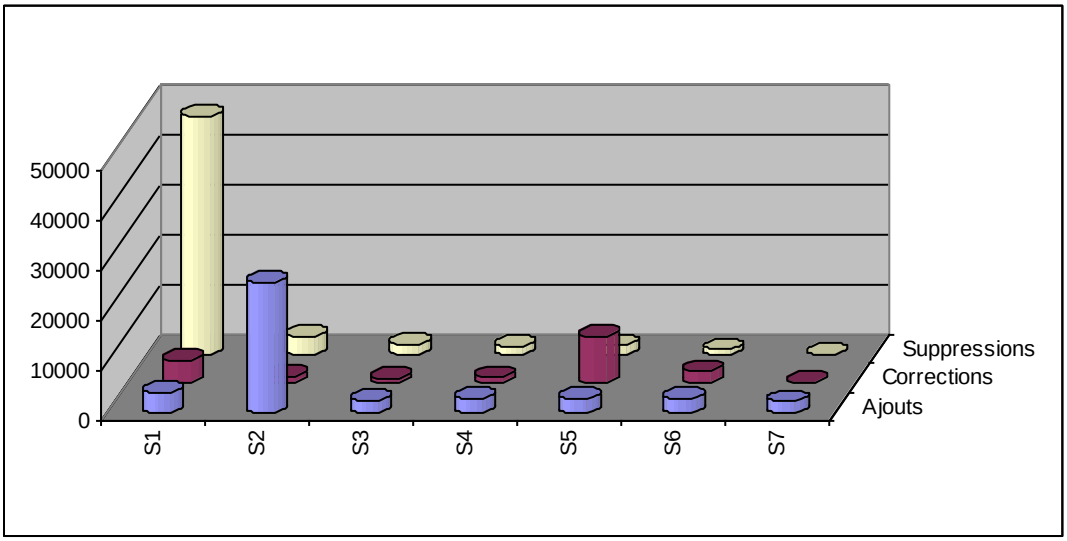


Figure 65 : Evolution du lexique morphosyntaxique anglais (septembre 2005 – mars 2009)

Les modifications de la première phase sont entièrement dues aux nouvelles spécifications que nous avons définies pour la détection des entités de noms géographiques (villes et pays) et d'entreprises. Ces deux traitements reposent sur des listes fermées qui ont été adaptées à ces nouvelles spécifications. Ainsi, les modifications du premier semestre sont le résultat de la réorganisation des noms géographiques, celles du deuxième semestre sont directement liées à l'ajout d'une liste de noms d'entreprises.

Les suppressions concernent une restriction de la reconnaissance des noms de villes, pour la plus grande partie françaises. Une bonne partie des corrections sont des ajouts d'étiquettes d'ambiguïté afin de tester une nouvelle stratégie mise en œuvre seulement en anglais pour

réduire le bruit dans la reconnaissance des noms géographiques fortement ambigus comme *Elisabeth*, *Charleston*, *Bell*, *Derbie*, etc. Les ajouts étant moins nombreux que les suppressions, il s'agissait bien d'une restriction de la détection. Dans les autres corrections on note l'ajout d'une étiquette marquant les noms qui se terminent en *ing* comme le gérondif. Les ajouts massifs ainsi que les suppressions du second semestre de cette première phase sont entièrement dus aux ajustements des noms d'entreprise.

L'augmentation des corrections marquant la deuxième phase concerne encore les entreprises, notamment le marquage d'ambiguïté des entreprises (ex. *Able Energy*). Les entreprises non ambiguës étaient marquées ambiguës alors que les entreprises ambiguës ne l'étaient pas. La plupart des suppressions et ajouts sont également destinés à améliorer la détection des noms d'entreprises. Pour la détection des noms géographiques, quelques suppressions ont eu lieu pour améliorer la recherche de ces mots quand ils font partie d'expressions (par exemple *Belgium* dans *Kingdom of Belgium*).

c. Le lexique allemand

L'évolution du lexique morphosyntaxique allemand est donnée dans le tableau 18 et illustrée dans la figure 66. Elle est globalement constante, avec environ 1000 à 2000 entrées ajoutées, corrigées ou modifiées par semestre, sauf pour deux phases successives d'ajouts et de corrections. Ce gain d'activité coïncide avec l'arrivée d'une personne supplémentaire au S4, dédiée à l'allemand, ce qui est vérifiable dans l'aperçu du nombre de commits par personne illustré dans la figure 62. La personne est marquée en jaune fluo. Cette activité retombe net avec le départ de la même personne.

	S1	S2	S3	S4	S5	S6	S7
Ajouts	1013	1919	7330	68170	1403	893	3
Corrections	498	2058	2034	2456	28362	145148	9
Suppressions	1095	293	4332	1053	866	2723	3
Total	2606	4270	13696	71679	30631	148764	15

Tableau 18 : Evolution du lexique morphosyntaxique allemand (septembre 2005 – mars 2009)

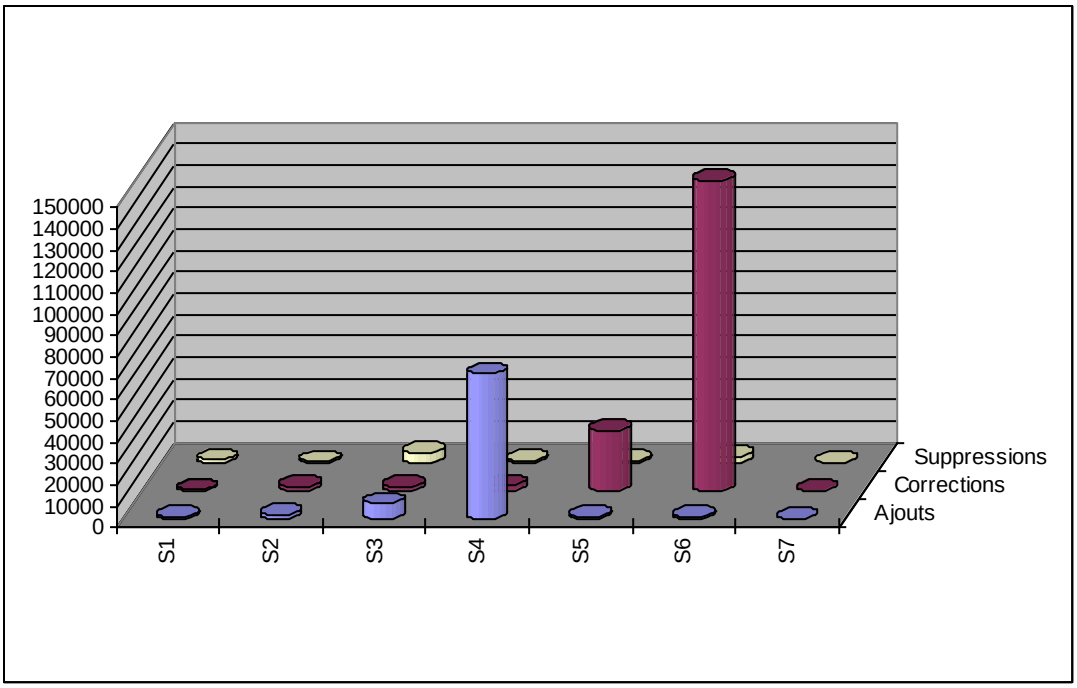


Figure 66 : Evolution du lexique morphosyntaxique allemand (septembre 2005 – mars 2009)

83% des ajouts du S3 et S4 sont des types 6 et 7, c'est-à-dire de nouvelles entrées dont ni le mot-forme ni le lemme n'existaient auparavant. Cette phase d'ajouts correspond majoritairement à un enrichissement en termes spécialisés et en noms d'entreprises. Les termes spécialisés ont été ajoutés pour répondre aux besoins d'un client dont le thésaurus devait être totalement couvert par le lexique. En même temps, les mots composés du même thésaurus devaient être bien décomposés. Certains de ceux-ci n'étaient en effet pas couverts par la grammaire de décomposition, comme par exemple les mots comprenant le composant *Öl* (huile) ou *Ei* (œuf). Les mots de deux lettres ont en effet été exclus des composants car ils conduisent le plus souvent à des décompositions sémantiquement incorrectes. Ainsi *Spielerei* (*bagatelle*) serait décomposé en *Spieler* et *Ei*, c'est-à-dire *œuf de joueur*. Les noms d'entreprise ont été rajoutés pour permettre la détection des noms d'entreprise. Nous avions en effet décidé de reconnaître seulement une liste fermée d'entreprises, en contexte pour celles dont le nom est ambigu. Le lexique joue donc un rôle central dans cette reconnaissance. A part ces deux types d'ajouts, un problème de génération de mots-formes équivalentes concernant le tréma (ä= ae, ö = ou, ü = ue) dans les mots composés a été corrigé. Cela a eu pour effet d'ajouter un nombre important de mots-formes équivalentes. Finalement, une petite partie des ajouts concernait des mots courants non présents dans le lexique, dont des professions (par ex. *Bibliothekar, Bildhauer*) qui servaient en même temps pour la détection des noms de personnes.

Cette phase d'ajouts a été immédiatement suivie d'une grande phase de corrections. Dans le S5, 7% des corrections étaient de type 2, soit des corrections du lemme. En effet une bonne partie des entrées ajoutées dans la phase précédente phase comportaient de mauvais lemmes. Le lemme de *Kathmandu* était par exemple *KathmanduKathmandu* au lieu du mot-forme même. Ces erreurs ont été introduites lorsqu'on a fait évoluer le codage du lemme, qui était auparavant calculé à partir du mot-forme. Les lemmes d'une partie des fichiers n'avaient pas été pris en compte, ce qui a provoqué un décalage temporaire avec le traitement. Celui-ci serait passé inaperçu si nous avions pris les versions des lexiques postérieures d'une semaine.

Les autres 93% de corrections étaient de type 1, c'est-à-dire un changement de la description. Pendant le S5, les étiquettes de la décomposition ont changé, entraînant 25 411 des 26 281 modifications de ce type comme nous l'avons constaté en vérifiant dans les lexiques. Sur le S6, toutes les descriptions des prénoms ont été modifiées par l'ajout d'une étiquette renforçant la prise en compte de la capitale du mot dans l'étiquetage. Les prénoms représentant 142 996 entrées, cette modification explique la quasi-totalité des corrections de type 1. Il convient ici de rappeler que le nombre d'entrées est directement lié au ratio entre mots-formes et lemmes comme nous l'avions illustré dans le tableau 31. Par exemple si on change un nom allemand, 8 entrées sont concernées, à cause des 4 cas au singulier et au pluriel. Les prénoms n'ayant pas de pluriel, il faut compter 4 entrées et non 8. Il ne s'agit donc pas de 142 996 prénoms, mais d'autant de mots-formes d'un peu moins de 30 000 prénoms. Ces deux corrections ont pu être réalisées automatiquement sans risque d'erreur et elles ne représentent même pas une heure de travail. Ainsi, si ces chiffres reflètent bien la vie du lexique, il faut les prendre avec précaution pour estimer les travaux linguistiques. Le reste des corrections a porté sur le marquage d'ambiguïté sur quelques prénoms (par ex. *An* qui est aussi une préposition, *Eden* ou encore *Jeans*, génitif de *Jean*) et le marquage de profession sur des entrées existantes.

Sans qu'on puisse parler d'un pic d'activité, le chiffre supérieur dans la série des suppressions correspond pour la plus grande partie à la suppression des noms géographiques comportant des abréviations (par exemple *a. = am, a.d.= an der*) en prévision d'un traitement à un autre niveau.

d. Le lexique espagnol

On identifie deux phases dans l'évolution du lexique espagnol. La première, S4, est marquée par des suppressions et un grand nombre de corrections. La seconde phase, S6-S7, démarre un an plus tard avec un grand nombre d'ajouts, de suppressions et de corrections dans le premier semestre, et des corrections qui continuent encore pendant le semestre suivant. Les deux phases estivales correspondent à des renforcements temporaires de l'équipe dédiés exclusivement à l'espagnol. La deuxième phase se termine avec l'embauche d'une personne permanente dans l'équipe.

	S1	S2	S3	S4	S5	S6	S7
Ajouts	1716	1351	41	1153	70	19154	217
Corrections	7	3	55	7905	0	7141	29576
Suppressions	0	92	399	44756	16	5686	480
Total	1723	1446	495	53814	86	31981	30273

Tableau 19 : Evolution du lexique morphosyntaxique espagnol (septembre 2005 – mars 2009)

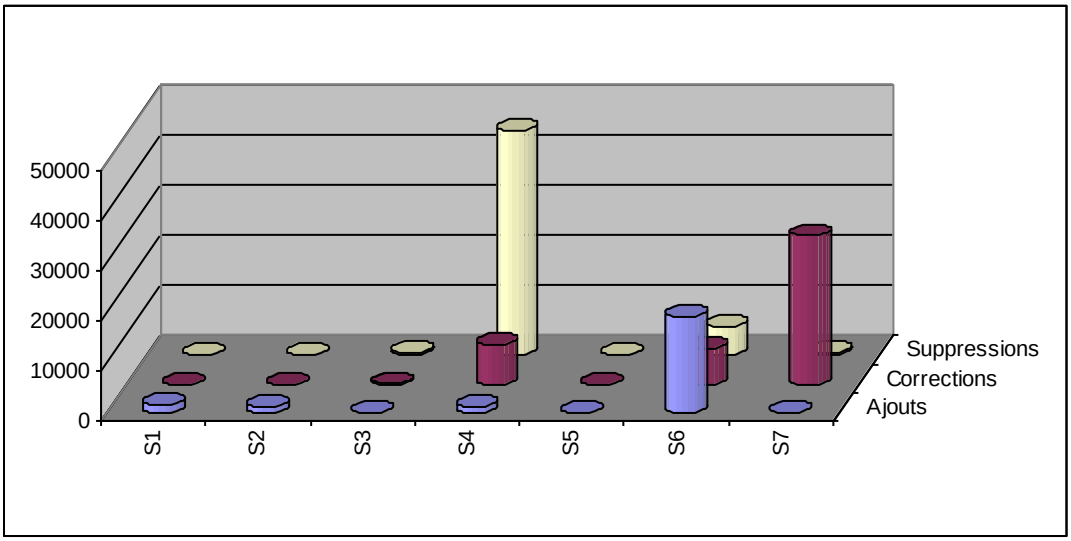


Figure 67 : Evolution du lexique morphosyntaxique espagnol (septembre 2005 – mars 2009)

Le grand nombre de suppressions de la 1e phase (S4) est dû au fait que les villes espagnoles n'ont pas été pris en compte suite aux nouvelles spécifications de détection de noms géographiques. Ce lexique contenait toutes les communes espagnoles, or seulement une sélection des communes basée sur le nombre d'habitants devrait être l'objet de ce traitement. La quasi-totalité des ajouts et des corrections sont liés aux entrées des prénoms, avec notamment l'ajout du marquage obligeant à tenir compte de la capitalisation dans l'étiquetage morphosyntaxique pour chaque prénom.

Le premier semestre de la seconde phase (S6) est totalement dédié à la mise à niveau de la détection des entités. Pour les noms géographiques on a ajouté des noms de villes espagnoles, pour les noms de personnes on a ajouté des étiquettes d'ambiguïté sur les prénoms et supprimé les prénoms composés, et pour les noms d'entreprises on a intégré une nouvelle liste d'entreprises, entraînant la suppression des anciennes entrées. Notons que la suppression des prénoms composés correspond à une nouvelle stratégie de reconnaissance : ceux-ci ont été remplacés par des règles dans la grammaire de détection des personnes. Le second semestre (S7) continue sur la lancée de l'amélioration de la détection de personnes, avec l'ajout d'un marquage sur les entrées de prénoms obligeant le respect total de l'accentuation à l'étiquetage et un marquage distinguant les prénoms espagnols des autres.

e. Le lexique italien

L'évolution du lexique italien sur la période considérée est caractérisée par une seule phase de modification, qui commence par un an de suppressions suivi d'un semestre d'ajouts et de corrections. Cette activité est essentiellement liée au renforcement temporaire de l'équipe pour mettre en place les détections des entités nommées selon des méthodes bien définies.

	S1	S2	S3	S4	S5	S6	S7
Ajouts	12	147	92	140	32	34 335	0
Corrections	6	1	9	52	6	25 988	7
Suppressions	142	6	136	4188	23 411	6110	14
Total	160	154	237	4380	23 449	66 433	21

Tableau 20 : Evolution du lexique morphosyntaxique italien (septembre 2005 – mars 2009)

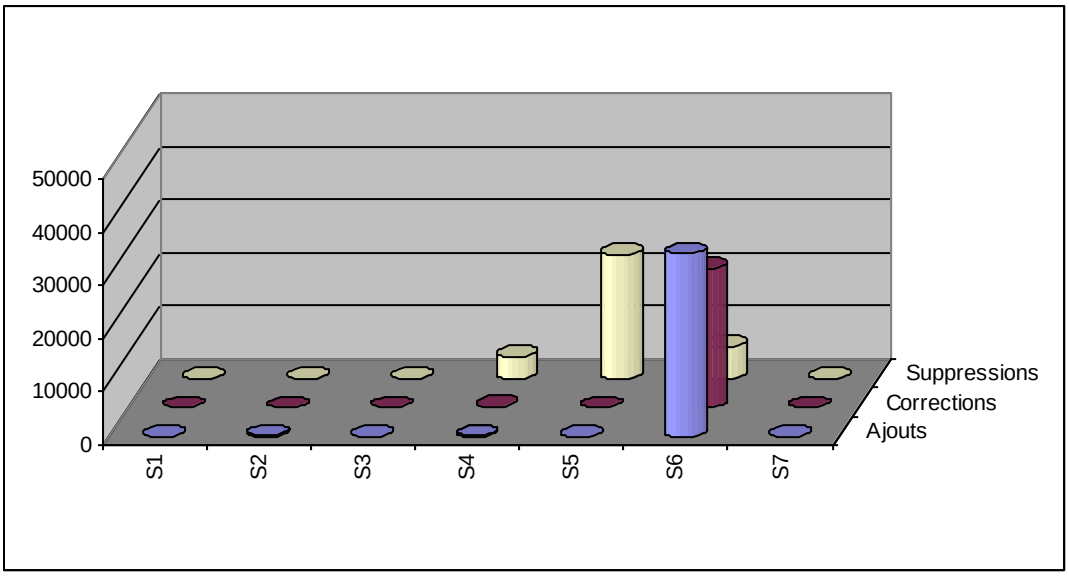


Figure 68 : Evolution du lexique morphosyntaxique italien (septembre 2005 – mars 2009)

Le premier semestre (S5) est marqué par des suppressions qui corrigent une surgénération d'entrées dans le paradigme d'un certain type d'adjectifs. Une ambiguïté erronée avait été ajoutée à la génération de ces entrées avec un fléchisseur. L'ambiguïté en question portait sur le genre du mot-forme au pluriel. Ainsi *gratuite* existait comme adjectif masculin et féminin pluriel, tandis qu'il est seulement féminin. En même temps, *gratuiti* existait comme masculin et féminin pluriel alors qu'il n'est en réalité que masculin.

Les modifications du deuxième semestre (S6) éliminent les adjectifs qui étaient également présents comme participes passés. Ce double codage était systématique, mais ne devrait pas l'être. Ceci correspond à une épineuse question linguistique à savoir dans quelles conditions les participes passés ont le droit d'une existence indépendante du verbe en tant qu'adjectif. Un critère sémantique peut être appliqué, mais demande un jugement au cas par cas et donc un travail conséquent.

Le dernier semestre (S6) de cette phase est marqué par l'ajout massif de noms d'entreprises pour le traitement de la détection des entreprises. Pour celle des noms de personnes, l'ajout d'un marquage d'ambiguïté sur les prénoms (par ex. *Adelaide*) est réalisé et les entrées de prénoms ont également été corrigées par l'ajout d'un marquage obligeant la prise en compte de la capitalisation lors de l'étiquetage morphosyntaxique. Cette dernière est également appliquée aux noms géographiques. La phase se termine avec la suppression des prénoms composés et de certains titres.

8.3.2.3 Malgré les divergences, une histoire commune

Si on peut avoir l'impression que les lexiques mènent leur propre vie, c'est loin d'être le cas. Les modifications sont sans exception liées aux traitements qui reposent sur elles. Dans la plupart des cas décrits ici, les lexiques sont modifiés suite aux changements dans les spécifications des traitements. Il est possible qu'ils entraînent de nombreuses modifications dans les lexiques en plus de celles des grammaires. Le nombre de modifications dans les lexiques varie en fonction de l'existence ou non de ressources dans la langue concernée. En allemand et en italien, la détection des entités était inexistante avant la période étudiée et aucun lexique de noms propres n'avait été intégré aux traitements. Cela explique pourquoi, dans l'adaptation pour atteindre les exigences des nouvelles spécifications, aucune

suppression n'a eu lieu dans ces lexiques, tandis qu'en anglais, français et espagnol, ces suppressions sont massives. En anglais et en français, les suppressions du premier semestre sont directement suivies d'un nombre élevé d'ajouts dans le semestre suivant, alors qu'en espagnol, ils arrivent le semestre d'après. Ces différences sont liées à la disponibilité de la ressource humaine. Traitement et ressource sont donc très liés : si le traitement est redéfini, les ressources doivent suivre.

A part quelques modifications liées à l'étiquetage morphosyntaxique, toutes sont liées à la détection des entités. Ce traitement étant construit sur le l'étiquetage, il est bien possible que l'erreur a été découverte lors de la mise en œuvre de la détection. Ainsi, en italien, au quatrième semestre, des corrections des paradigmes de flexion ont eu lieu pour un certain type d'adjectifs. L'entrée de nos lexiques étant le mot-forme et la classe de flexion n'ayant pas été préservée quand le lexique a été développé, les corrections demandent un peu d'ingéniosité : il faut d'abord retrouver toutes les entrées concernées avant d'appliquer les modifications. La correction aurait été plus facile si le lexique était géré par des couples *<lemme ; classe de flexion>* : au lieu de corriger tant d'entrées, seules les règles correspondant à la classe de flexion auraient à être corrigées, c'est-à-dire deux modifications au lieu de plus de 4000. Cette façon de gérer éviterait aussi des erreurs de lemmes mal codés : cette information est redondante sur toutes les entrées, car les entrées sont les mots-formes.

Les modifications lexicales liées à l'étiquetage morphosyntaxique ne sont d'ailleurs pas toujours évidentes à trancher. A l'origine se trouve la friction entre théorie linguistique et traitement automatique du langage. Si les critères d'analyse linguistique semblent clairs en théorie, en pratique l'analyse automatique n'est pas toujours aussi simple. Ainsi, en néerlandais et en allemand, tous les infinitifs peuvent être utilisés comme noms, tout comme le gérondif en anglais. Pour autant, quelle doit être l'analyse morphosyntaxique dans ces deux cas ? Le codage linguistique découle de la réponse à cette question : soit on met les deux codages, soit un seul. Pour le moteur de recherche de Sinequa, c'est important, car les mots sont normalisés à leurs lemmes pour améliorer à la fois précision et rappel. Nous avons tranché que la catégorie du mot est inaliénable. Un verbe reste donc un verbe, même s'il peut être utilisé en position nominale, avec évidemment toutes les conséquences pour le modèle de langage d'étiquetage morphosyntaxique. D'autres questions du même type concernent la différence entre un participe passé et un adjectif, le codage des noms qui peuvent toujours être adjectifs ou inversement, les critères pour distinguer les mots composés de collocations fortuites. Dans ces cas, le pragmatisme l'emporte en laissant la visée industrielle avoir le dernier mot pour trancher des questions épineuses. L'application finale a donc une influence directe sur le codage des ressources.

Les lexiques portent aussi un héritage historique qui peut augmenter le nombre de modifications. Cet historique peut être récent et lié aux expériences ou à des adaptations prématurées, auquel cas les modifications prennent la forme d'une suppression massive suivie d'un grand ajout. Les modifications en français et en anglais sont pour des raisons commerciales prioritaires sur les autres langues. Les méthodes sont donc d'abord mises au point dans ces deux langues, avant d'être relayées sur les autres. Comme les méthodes à appliquer ne sont pas toujours stables au début, il est possible qu'on revienne en arrière sur certaines modifications, ajoutant un surplus de modifications. Le grand nombre de suppressions en anglais et en français au 1er semestre et en espagnol au 4^{e} semestre suivi d'un nombre élevé d'ajouts dans le semestre ou la période suivants en sont de bons exemples. En pourcentages, ces suppressions et ajouts prennent une bonne part dans les modifications comme nous pouvons voir dans la Figure 69.

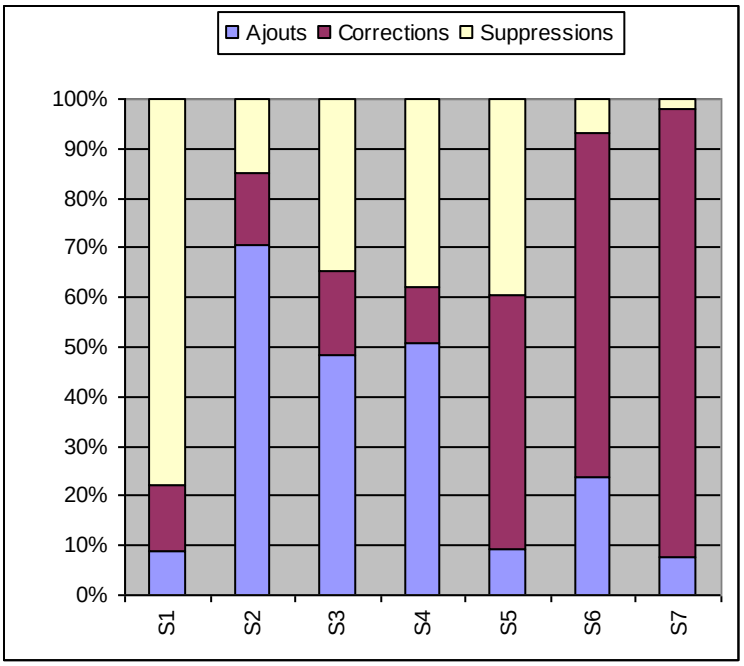


Figure 69 : Pourcentages des différents types de modification selon la période toutes langues confondues

Cet historique peut aussi être plus vieux, et être lié à un traitement tombé aujourd'hui en désuétude, ce qui se traduit en général par des suppressions et des corrections. Ainsi, dans notre lexique du français se trouvent un bon nombre d'entrées et d'étiquettes qui sont dédiées à la correction orthographique du temps où les mêmes lexiques étaient utilisés dans *Voltaire*[111]. Ces entrées peuvent être gênantes, car elles sont sous-spécifiées (par exemple *Pompidou* est codé comme un nom propre mais pas comme un nom de famille) ou sont en contradiction avec le traitement (aucun nom de famille étant codé, *Pompidou*, aussi célèbre soit-il, ne devrait pas faire exception à la règle).

L'importance accordée à l'anglais et au français ainsi que les caractéristiques des entrées modifiées expliquent l'évolution identique des lexiques de ces deux langues comme on peut le voir sur la figure 70. Nous y représentons le total des modifications par langue sur la période étudiée. Le grand nombre de modifications sur le français et l'anglais aux deux premiers semestres s'explique par la réécriture des spécifications de la détection d'entités juste avant la période étudiée. Les modifications dans les autres langues sont plus éparses et beaucoup plus massives sur un laps de temps plus court. Même si c'est moins flagrant, leur évolution est aussi très semblable.

[111] *Voltaire* était le correcteur orthographique commercialisé par CORA, le prédécesseur de Sinequa, dans les années 90.

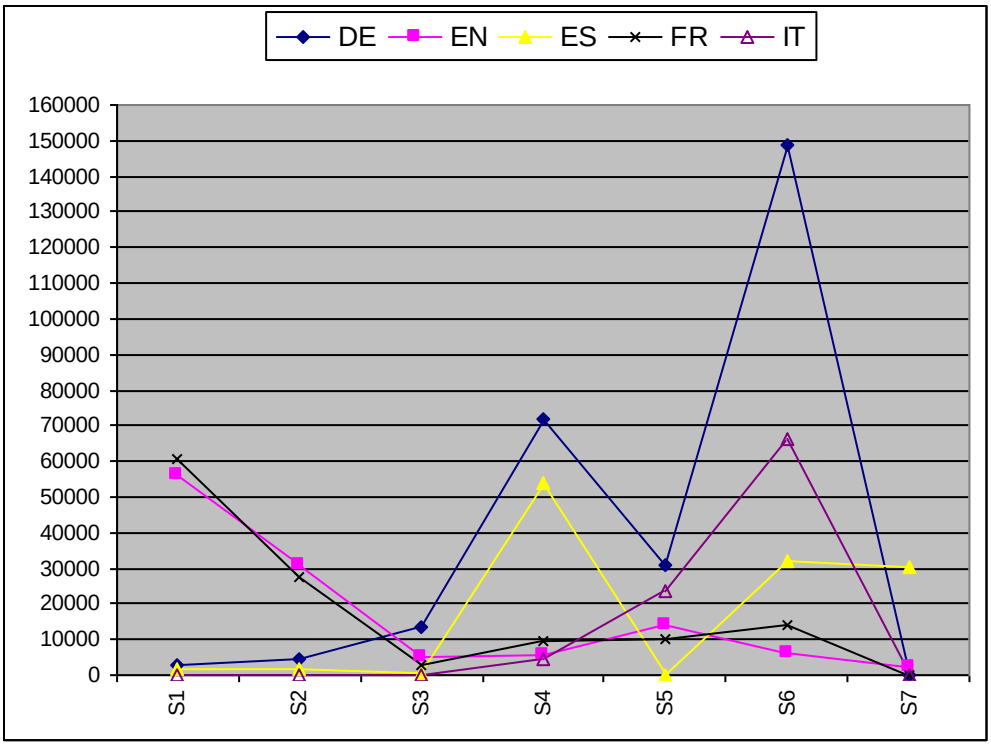


Figure 70 : Evolution des lexiques selon le nombre de modifications par période

Nous avons donc deux groupes de lexiques, l'un avec une évolution identique, l'autre avec une évolution similaire. Tous les lexiques doivent se conformer aux mêmes spécifications, seulement, le français et l'anglais ont priorité, et les ressources humaines pour les autres langues ne sont que périodiquement disponibles. Les méthodes sont donc testées et éprouvées en français et en anglais avant leur portage dans les autres langues. Ces modifications profitent donc de l'expérience gagnée et sont exécutées sur un court laps de temps, ce qui est aussi lié à la disponibilité des ressources humaines. Celle-ci se concentre aux semestres d'été pour la raison que c'est la période de prédilection pour les stages. Un encadrement strict permet alors d'aller assez vite dans l'encodage des ressources. C'est d'autant plus vrai que certaines ressources ont été centralisées pour avoir une plus grande consistance et une gestion plus facile entre les différentes langues. Après moult modifications en français et en anglais, les noms d'entreprises et les prénoms de toutes les langues ont été mis en commun afin de constituer une base unique pour les différentes langues. Nous avions déjà centralisé la gestion des noms géographiques avant la période étudiée. Cette centralisation des noms propres a le grand avantage de rendre l'implémentation bien plus rapide dans une nouvelle langue. Les abréviations supprimées car trop ambiguës et les prénoms composés traités à un autre niveau sont autant d'exemples de décisions prises suite à des expériences en français et en anglais qu'il ne fallait qu'exécuter dans les autres langues selon le modèle déjà établi.

8.3.2.4 Le suivi des modifications ne traduit pas le travail du linguiste

Dans la méthode que nous avons conçue pour faire les calculs de modifications, nous avons remarqué que les corrections cachent parfois des ajouts d'informations. L'ajout d'un marquage morphosyntaxique ou sémantique dans une entrée existante était interprété comme une correction, alors qu'on pourrait l'interpréter comme un ajout. Pour cela il faudrait non pas considérer la description comme un bloc d'information, mais prendre en compte chaque

information comme un élément. A notre avis, cela aurait considérablement complexifié le calcul sans pour autant apporter d'informations importantes.

La figure 71 montre les types de modifications (voir tableau 13, p. 185) selon la période, avec *a* pour ajout, *c* pour correction. Nous voyons que quatre types concentrent les modifications. Il s'agit de la suppression (violet), de la correction de type 1 (bleu pale) et de l'ajout de type 7 (bleu marine) ou de type 6 (rouge pale). Ces quatre types correspondent respectivement aux suppressions, aux corrections de la description (ce qui couvre aussi les ajouts dans la description), aux ajouts dont la description est complètement nouvelle, et aux ajouts dont la description existait déjà.

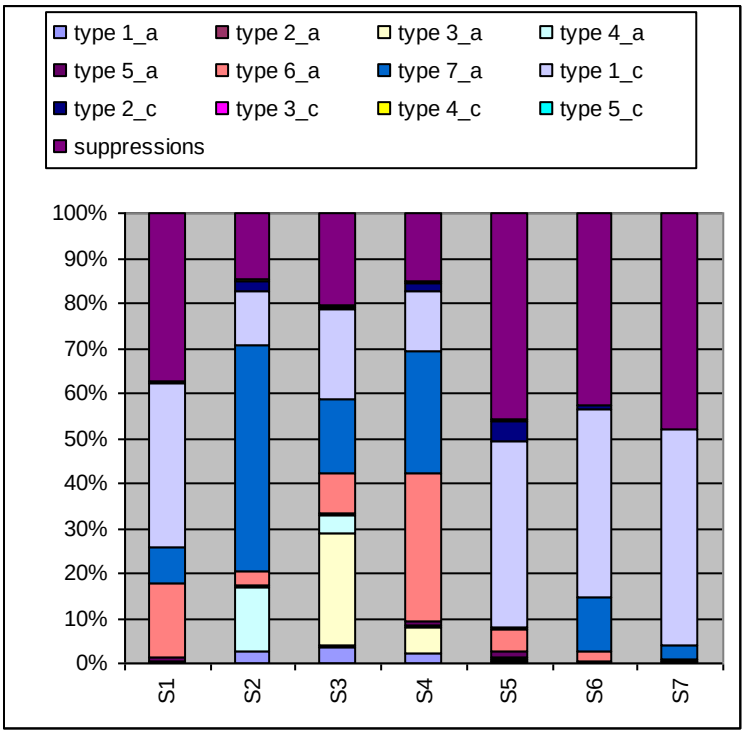


Figure 71 : Pourcentage des types de modification selon la période

Les calculs sans leur mise en contexte ne suffisent pas pour apprécier le travail fait par le linguiste, car les modifications ne représentent pas tous la même charge de travail : certaines modifications peuvent être très longues, car elles demandent un jugement linguistique, alors que d'autres peuvent être massives mais rapides car réalisables automatiquement. Dans la détection d'entités, les modifications de grammaires sont au moins aussi importantes, mais nous n'avons pas trouvé de moyen de calculer leur évolution.

Pour calculer ces résultats, nous avons fait abstraction de la réalité lexicale, c'est-à-dire de la répartition des entrées lexicales sur plusieurs lexiques physiques. Si nous l'appliquions, nous verrions des résultats tels que dans le tableau 21, où A et B représentent des ensembles de lexiques, respectivement de langue générale et des noms d'entreprise. La séparation de ces deux ensembles existe pour des raisons d'optimisation, le lexique B étant exclu du lexique si l'extraction des entités ne fait pas partie de la vente conclue, entraînant une réduction des entrées lexicales et allégeant ainsi les calculs et la mémoire des processus. Les chiffres montrent le nombre d'ajouts et de suppressions entre deux versions des lexiques A et B à six mois d'intervalle (fins de S4 et de S5). Comme nous allons voir, les 2210 ajouts de B

proviennent pour la plupart de A, où l'on compte 2881 suppressions. B étant vide auparavant, on n'y compte aucune suppression.

	A	B	Total
Ajouts	1208	2210	1403
Suppressions	2881	0	866

Tableau 21 : Ajouts et suppressions en cas de transfert intralexical

Au final, on compte 1403 ajouts. Ce nombre n'est pas la somme des ajouts de A et de B, car il a en partie été neutralisé par des suppressions en A. On voit le même phénomène dans les suppressions, où les suppressions au total ne sont pas la somme des suppressions de A et de B. Nous calculons le nombre d'ajouts provenant du transfert vers B en deux temps :

$$Total(ajouts) - ajouts(A) = vrais\_ajouts(B)$$

$$Ajouts(B) - vrais\_ajouts(B) = entrées\_transférées\_de\_A\_vers\_B$$

Nous soustrayons donc du nombre d'ajouts de B la différence du nombre d'ajouts au total et des ajouts de A : 2210-(1403-1208)= 2015. Le résultat est le nombre de suppressions de A qui a été transféré, et donc ajouté à B. Ce calcul se vérifie : les 866 suppressions au total égalent la différence entre les 2881 suppressions de A et 2015.

Le niveau d'abstraction que nous avons choisi pour calculer l'évolution a l'avantage de montrer une image globale tous lexiques confondus. Néanmoins, si nous voulions vraiment faire un suivi détaillé des activités linguistiques, il faudrait savoir tracer les transferts interlexicaux. Ceux-ci existent dès que l'on crée un lexique spécialisé à partir d'un lexique existant tout en supprimant les entrées dans le lexique d'origine, ce qui peut présenter un travail considérable. Les informations que nous avons à disposition ne permettent pas de faire ce genre de suivi. Pour que ce soit possible, il faudrait mettre en place un système de gestion qui trace toutes les modifications. Pour rendre ceci possible, il faudrait abolir la séparation physique qui existe aujourd'hui entre les lexiques. Elle existe pour des raisons pratiques de gestion, un lexique de prenoms par exemple étant plus facile à gérer s'il n'est pas mélangé à un lexique de langue générale. Nous préconiserions donc un modèle centralisé où des vues correspondraient à ce genre de besoin.

Que nous apportent donc ces calculs pour la gestion ? Ils nous procurent une vision très haut-niveau sur l'évolution du lexique, qui était totalement absente. Les analyses n'avaient jamais été faites auparavant, et ces évolutions étaient assez obscures. Pour suivre le travail fait par les linguistes, les résultats présentés ne suffisent cependant pas, car ils ne sont pas interprétables automatiquement sans regarder en détail les modifications, d'autant plus que ajouts et suppressions peuvent s'annuler et être comptabilisés ou pas selon la période choisie.

8.4 Conclusion

Nous avons présenté les outils que nous avons mis en place destinés à procurer une vision sur l'état et l'évolution des ressources. Nous avons travaillé sur la quantification des informations lexicales et leur comparabilité entre les langues, la complexité des grammaires codées sous forme d'automates et l'évolution des ressources lexicales. Les deux premiers donnent des informations sur l'état des ressources, alors que le dernier permet de suivre leur évolution en termes de modifications. Les méthodes mises en œuvre dans ce chapitre sont simples mais originales, n'ayant pas trouvé de point de comparaison dans la littérature scientifique.

Le suivi de l'évolution des lexiques s'est fait en comparant les versions[112] des lexiques avec des intervalles d'un semestre. Nous avons appliqué une heuristique pour déterminer si les modifications effectuées étaient des ajouts, des corrections ou des suppressions. L'analyse de ce suivi sur les cinq langues principales du moteur donne des informations sur la façon dont les lexiques sont gérés. La principale conclusion est que le français et l'anglais servent de langues d'expérimentation, et lorsque les méthodes se sont affinées, les modifications sont répercutées sur les ressources des autres langues.

Nous avons également démontré qu'il faut résister à la tentation de prendre les chiffres fournis par le système pour quantifier le travail d'un linguiste. Aucun chiffre fourni ne nous a procuré une bonne vision sur le travail réalisé. Nos outils ne suffisent donc pas pour cela, mais si on estime vraiment nécessaire de recourir à des outils pour estimer le travail des linguistes, nous sommes d'avis qu'il faut plutôt revoir le suivi des travaux qu'on leur confie.

[112] Nous avons la trace des versions grâce au système de *versioning* que nous avions mis en place précédemment (voir 7.1.1, p. 147)

Chapitre 9

Accélérer l'acquisition de ressources lexicales

L'acquisition des ressources est une opération qui a la réputation d'être laborieuse en raison de la quantité et la complexité des informations. Un environnement de gestion doit fournir les outils pour accélérer et faciliter cette opération. Nous nous concentrons sur le lexique morphosyntaxique qui rassemble la majeure partie des informations dans beaucoup de cas. Ces outils sont intimement liés au format de codage : le choix de coder le lexique en extension ou en intension détermine aussi la façon de le gérer. Alors que notre méthode d'acquisition habituelle privilégie l'intégration de ressources existantes en extension, nous avons pu mettre en œuvre la gestion par intension pour le lexique finnois. Le codage en intension repose sur l'utilisation active de paradigmes de flexion. Dans ce chapitre nous commençons par introduire le concept du paradigme de flexion, tout en montrant ses limites. Nous présentons ensuite l'exemple du finnois qui a été un cas d'acquisition tellement atypique qu'il nous a obligé à adopter une vision différente de la gestion d'un lexique de grande couverture. Pour accélérer l'acquisition d'informations lexicales, nous avons développé un outil expérimental que nous avons testé sur l'allemand. Lors de ces travaux, nous avons dressé une typologie des erreurs rencontrées.

9.1 Extension ou intension ?

9.1.1 L'architecture lexicale extensionnelle par défaut

L'architecture lexicale est une notion que nous avons uniquement trouvée dans [Sérasset, 1994], qui la définit dans le cadre de l'organisation de ressources multilingues :

> « L'architecture lexicale d'une base lexicale définit l'ensemble des dictionnaires qu'elle contient, leurs types et leurs relations. » (p. 57). « Cette architecture lexicale définit les dictionnaires de la base et le type de chacun. Ces dictionnaires peuvent être monolingues, bilingues ou interlingues. » (p.67).

Nous redéfinissons cette notion d'architecture lexicale en la généralisant et en introduisant l'existence de règles ainsi qu'une référence au système pour lequel l'architecture a été conçue :

> L'architecture lexicale d'un système est le modèle sous-jacent qui définit l'interaction entre lexiques, règles de flexion et règles d'analyse lexicale.

Nous distinguons deux architectures lexicales : en extension ou extensionnelle et en intension ou intensionnelle. Les mots-formes des lexiques en extension sont tous listés, alors que les lexiques en intension sont organisés en bases et classes de flexions. Cette dernière architecture nécessite alors une opération supplémentaire impliquant les paradigmes de flexion correspondants avant que le lexique puisse s'utiliser pour le traitement de textes.

Le principe d'une architecture basée sur les paradigmes de flexion n'est pas nouveau. Parmi des lexiques organisés de cette façon, nous pouvons mentionner les lexiques du LADL ([Courtois, 1990]), le lexique morphologique croate HML ([Tadic et Fulgosi, 2003]), et le lexique français décrit dans [Loupy et Gonçalves, 2008]. Si ce type d'organisation intensionnelle semble plus utile pour les langues comptant beaucoup de flexions (le croate compte sept cas) afin de réduire la masse de données à gérer, elle peut s'appliquer à toute langue flexionnelle comme bonne pratique de gestion lexicale.

L'architecture lexicale que nous retrouvons à Sinequa est extensionnelle, sauf pour le finnois. Le choix de ce type d'architecture est lié à la méthode d'acquisition déployée[113]. Si les ressources sont parfois collectées sur le web, elles proviennent en général de laboratoires universitaires qui disposent déjà de ressources lexicales étendues. Une collaboration est initiée garantissant l'adaptation de la ressource. Cette méthode semble la plus rapide pour mettre en place un lexique. Pour les mots à ajouter, des fléchisseurs sont développés pour générer les formes fléchies et leurs informations lexicales à partir de lemmes et de leur classe de flexion. Excepté pour le finnois, tous les lexiques à large couverture des langues dont nous avons pu encadrer le développement ont été construits à partir de ressources existantes, en les adaptant et en les enrichissant.

[113] Cette méthode est décrite en Annexe H. La première étape de recherche de ressources étant souvent entravée par des descriptions incomplètes des ressources, nous proposons d'inclure dans la description des ressources les informations listées en Annexe I.

9.1.2 Le lexique finnois codé en intension

La publication fin 2006 du lexique du *Research Institute for the Languages of Finland* étant postérieure de quelques années au développement du finnois chez Sinequa, notre recensement initial mentionnait seulement le GREYC comme laboratoire disposant de ressources électroniques finnoises, mais nous n'avions pas pu nous accorder sur une collaboration.

Le lexique du finnois est donc codé en intension. Les règles étant trop nombreuses pour être codées efficacement dans le programme même, la grammaire de flexion est séparée du programme exploitant ces règles. Etant donné le grand nombre de règles, il fallait en effet pouvoir intervenir sur la grammaire sans recourir à un informaticien.

L'architecture lexicale du finnois est particulière par rapport aux autres langues. Ceci est lié d'une part aux ressources disponibles pour construire le lexique, et d'autre part aux propriétés de la langue.

Comme le hongrois, le finnois est une langue agglutinante. S'y ajoutent les phénomènes d'harmonie vocalique et d'alternance consonantique qui ont des conséquences sur les lois de la construction morphologique des mots. Pour mieux comprendre le défi du traitement de cette langue, nous expliquons en quelques mots ses particularités, afin de mieux comprendre nos choix de traitement. Par harmonie vocalique, on entend le fait que les voyelles a, o et u ne peuvent coexister dans un mot non composé avec les voyelles ä, ö et y. Le i et e restent neutres, mais en l'absence d'autres voyelles, le mot prend les terminaisons dans la série des ä, ö et y. L'alternance consonantique spécifie que, pour chaque base de mot, il existe un degré fort et un degré faible, dont l'alternance dépend de l'ouverture ou fermeture de la syllabe suivante. Pour *kompa*, le nominatif singulier (N.S) est au degré fort alors que les deux autres cas de référence, le génitif singulier (G.S) et l'illatif pluriel (IT.P) du même mot, sont au degré faible. La transformation sur la base est mp > mm.

FI	rampa	N.S		ramman	G.S	rampoihin	IT.P
FI	lämpö	N.S		lämmön	G.S	lämpöihin	IT.P
FI	mesi	N.S		meden	G.S	messiin	IT.P
FI	kompa	N.S		komman	G.S	kommissa	IT.P

Ces contraintes phonétiques sont tellement fortes que les exceptions sont très rares (mais existent dans des emprunts).

Le système d'analyse du finnois, tel que nous l'avons conçu pour le moteur de recherche, consiste en un lexique de mots-formes irrégulières, un lexique de bases morphologiques et un ensemble de règles de flexion. Nous distinguons une trentaine de classes de flexion en ignorant les deux phénomènes dont nous parlions ci-dessus. Une classe de flexion est associée à chaque base, et un fléchisseur interprète la base et la classe pour générer les mots-formes du paradigme de flexion et leurs descriptions. A cette trentaine de classes correspondent 19 230 lignes de terminaisons, avec une règle par ligne. 8 200 d'entre elles sont réservées à la seule flexion des participes d'agent. 38 paradigmes de flexion existent pour les verbes, 64 pour les noms et adjectifs. Tous ces paradigmes se simplifient respectivement sur 9 et 20 classes de flexion grâce à des filtres sur les motifs des mots tenant compte du motif du degré et de l'harmonie vocalique.

Certains paradigmes sont très verbeux, comme celui du verbe participe d'agent, un sous-paradigme du verbe qui compte 200 mots-formes. La grande régularité du finnois cautionne cette multitude de mots-formes : la déclinaison des verbes n'a pas d'exception sauf pour quelques formes fléchies du verbe *olla* (être). La conséquence immédiate est qu'il existe peu d'ambiguïté morphologique et sémantique. Les règles n'ont pas été intégrées directement

dans l'analyseur, ce qui empêche de mettre au point un devin pour les unités textuelles hors vocabulaire. Sur l'axe des systèmes à lexique ou à règles (figure 15, p. 96), cela place notre système pour le finnois plutôt à droite du centre. Si on intégrait toutes les règles pour une reconnaissance totalement dynamique, il se trouverait à l'extrême droite de l'axe.

L'architecture lexicale de notre système d'analyse morphosyntaxique du finnois est illustrée dans la figure 72. L'exemple donné est *ymmärtäminen* (compréhension, intelligence). Le lexique de bases et la grammaire sont activement entretenus. En cas de modification, le lexique de mots-formes est regénéré, fournissant ainsi des millions de mots-formes. A Sinequa, cette architecture est unique, car pour les autres langues, le lexique de bases morphologiques et la grammaire de flexion n'existent pas.

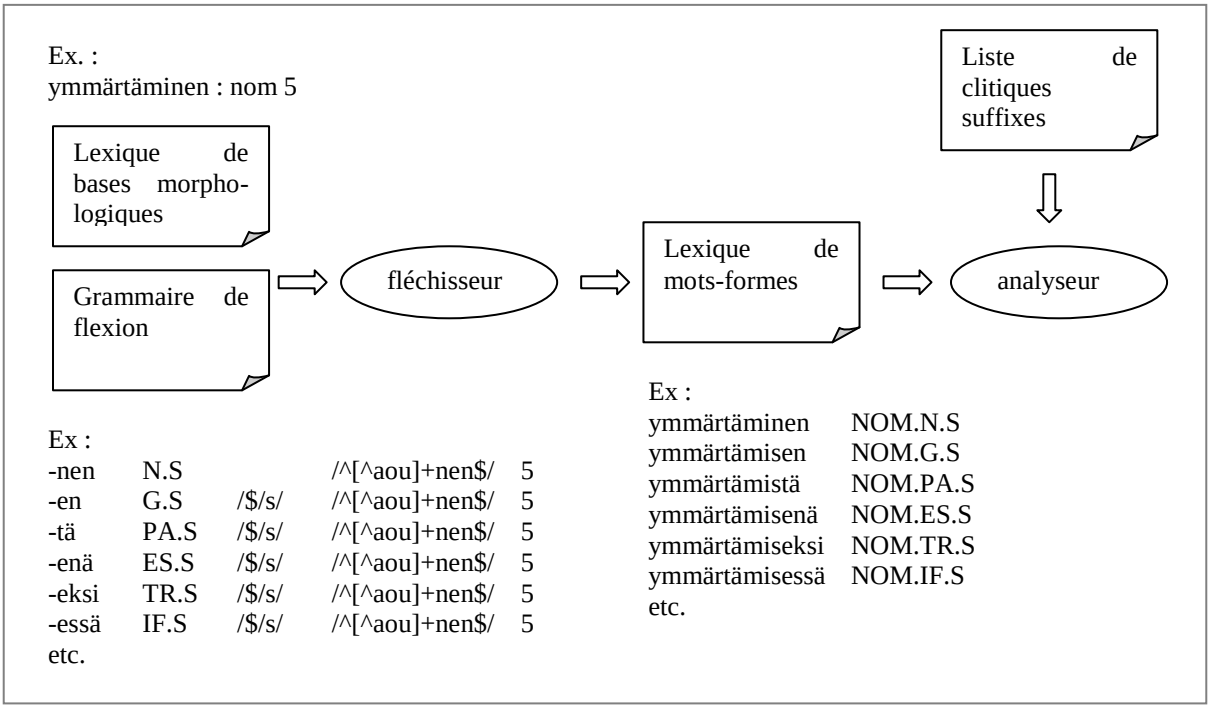


Figure 72 : Architecture lexicale du système d'analyse morphosyntaxique du finnois

Nous sommes passé par l'étape intermédiaire d'un lexique de mots-formes pour le finnois pour être conforme au cadre des procédures existantes. Celles-ci compilent les données sous forme binaire pour exploitation par l'analyseur. Si l'analyseur savait interpréter la grammaire de flexion et l'appliquer sur le lexique de bases morphologiques, nous aurions pu nous passer du fléchisseur et de la génération du lexique de mots-formes. Cette architecture lexicale est idéale quand on dispose d'un lexique intensionnel (figure 73). L'analyseur reconstitue les mots-formes à partir des informations lexicales ou bien analyse les unités textuelles en appliquant les règles. Du point de vue de la gestion, il est séduisant de disposer d'un lexique réduit en nombre d'entrées : la probabilité d'introduire des erreurs se réduit avec le volume de données.

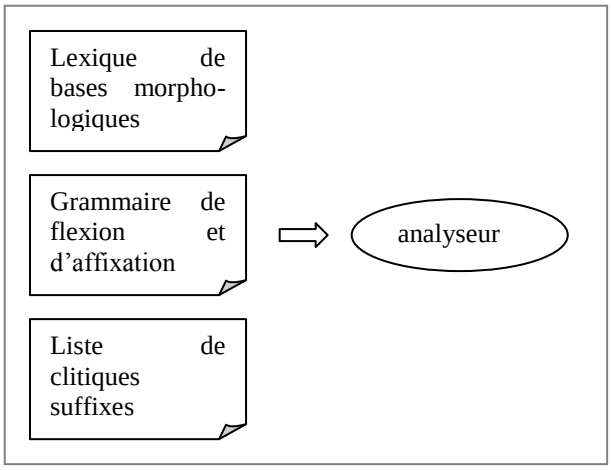


Figure 73 : Architecture lexicale simplifiée

Il faut néanmoins admettre que l'écriture de la grammaire de flexion et la classification des entrées ne sont pas toujours évidentes. La grammaire peut contenir un grand nombre de classes de flexion, et celles-ci ne correspondent pas toujours aux classes décrites dans les grammaires traditionnelles. Tout comme nous avons réduit les 102 classes de flexion en finnois à 29, il faut regrouper les classes de flexion par des super-classes qui réduire leur nombre et indiquer les critères pour faire la distinction automatiquement à l'intérieur des super-classes. La réduction du nombre de classes est importante pour que les personnes qui les manipulent puissent les mémoriser facilement.

A moins d'arriver à une sorte de normalisation des classes et des super-classes, tout lexique constitué de couples <lemmes, classe de flexion> restera intimement lié à l'application contenant les règles. Le partage d'un lexique de ce type n'est donc pas très utile sans publier également les règles de flexion qui l'accompagnent.

9.1.3 Les limites de l'architecture lexicale en intension

Le nombre de mots-formes par paradigme flexionnel n'est pas une donnée absolue. Il dépend des différents facteurs abordés dans cette section. Certains sont liés à la langue, d'autres à l'application ou au choix de l'encodeur.

Le nombre de mots-formes d'un paradigme dépend tout d'abord de la catégorie du paradigme : en français, les paradigmes des verbes contiennent bien plus de mots-formes que ceux des noms.

Il varie ensuite selon les propriétés de chaque langue : les paradigmes verbaux anglais contiennent moins de mots-formes que les français. Les langues à cas, comme l'allemand et le russe contiennent plus de mots-formes par nom que le français ou l'anglais (voir les exemples en 2.2.2).

```
FR      Pf  (axiome, nom) = {
                    (axiome, nom),
                    (axiomes, nom)
                    }
```

DE Pf (Axiom, nom) = {
(Axiom, nom),
(Axioms, nom),
(Axiomes, nom),
(Axiomen, nom),
(Axiome, nom)
}

Le nombre d'entrées dépend aussi de la granularité de la description. Avec une description plus fine, le paradigme contient plus de mots-formes en allemand, *Axiom* et de *Axiome* étant ambigus, alors qu'en français, le nombre d'entrées reste identique. L'ajout du genre, du nombre et du cas dans la description a introduit cette ambiguïté.

FR Pf (axiome, nom masculin) = {
(axiome, nom masculin singulier),
(axiomes, nom masculin pluriel)
}

DE Pf (Axiom, nom neutre) = {
(Axiom, nom neutre singulier au nominatif),
(Axiom, nom neutre singulier à l'accusatif),
(Axiom, nom neutre singulier au datif),
(Axioms, nom neutre singulier au génitif),
(Axiomes, nom neutre singulier au génitif),
(Axiomen, nom neutre pluriel au génitif),
(Axiome, nom neutre pluriel au nominatif),
(Axiome, nom neutre pluriel à l'accusatif),
(Axiome, nom neutre pluriel au génitif)
}

Dans les descriptions également, il y a une répétition d'informations, notamment de la catégorie grammaticale et du genre, mais aussi du nombre et du cas. Il est alors possible de factoriser la représentation en faisant hériter les parties de descriptions communes de la description plus générale, comme la catégorie grammaticale et le genre[114]. Le nombre de mots-formes dans le paradigme flexionnel dépend donc également de la factorisation appliquée, même s'il existe une équivalence logique.

Une des factorisations possibles du paradigme illustré plus haut est représentée ci-après. Sa représentation équivaut à la précédente, et d'autres sont possibles.

DE Pf (Axiom, nom neutre) = {
(Axiom, {singulier, {au nominatif, à l'accusatif, au datif}}),
(Axioms, {singulier au génitif}),
(Axiomes, {singulier au génitif}),
(Axiomen, {pluriel au datif}),
(Axiome, {pluriel, {au nominatif, à l'accusatif, au génitif}})
}

Le nombre de mots-formes par paradigme dépend aussi du point de vue linguistique. Par exemple, les adjectifs en allemand peuvent être décrits par une description mentionnant le cas,

[114] L'héritage du genre n'est pas universel : il existe des exceptions comme *ciglio* en italien (voir les exemples en 5.4).

comme pour les noms, ou bien par une description de la désinence. Dans le premier cas, l'ambiguïté morphosyntaxique est omniprésente, dans le deuxième elle est inexistante.

```
DE      Pf (lieb, adjectif) = {
                (lieb,    {singulier {nominatif, {masculin, neutre}})
                (liebe,   {singulier {nominatif féminin, accusatif {masculin, féminin,
                          neutre}}, pluriel {nominatif, accusatif})
                (lieben {singulier {accusatif masculin}, pluriel datif})
                (liebes {singulier {accusatif neutre, génitif {masculin, neutre}})
                (lieber {singulier, {nominatif masculin, génitif féminin}, pluriel
                          génitif}
                }

DE      Pf (lieb, adjectif) = {
                (lieb,    0)
                (liebe,   E)
                (lieben EN)
                (liebes ES)
                (lieber ER}
```

Il va de soi qu'un tel choix a de grandes conséquences sur l'écriture de règles d'analyse syntaxique. Dans le premier cas, on peut imaginer une fonction de *cas identique* pour des mots successifs comme un simple test de comparaison, mais dans le deuxième cas, il faut décrire les combinaisons entre terminaisons et cas. La simplification obtenue dans le lexique demande ainsi en contrepartie une plus grande complexité au niveau des règles syntaxiques.

Un exemple similaire en français est celui du choix entre singulier et pluriel ou *singulierpluriel* pour les mots qui ont les mêmes mots-formes au singulier et au pluriel. Dans une description où on privilégie l'absence d'ambiguïté, on décrira les mots-formes comme singulier/pluriel plutôt que de laisser une ambiguïté singulier et pluriel. L'exemple est un mot construit, mais nous aurions aussi pu prendre n'importe quel adjectif se terminant en *–eux*.

```
FR      Pf (anti-rides, nom masculin) = {
                (anti-rides,      {singulier, pluriel})
                }

FR      Pf (anti-rides, nom masculin) = {
                (anti-rides,      singulierpluriel)
                }
```

Le même problème se pose avec les mots-formes ambigüs au masculin et au féminin.

```
FR      Pf (paléolithique, adjectif) = {
                (paléolithique,           singulier, {masculin, féminin})
                (paléolithiques,          pluriel, {masculin, féminin})
                }

FR      Pf (paléolithique, adjectif) = {
                (paléolithique,           singulier, masculinféminin)
                (paléolithiques,          pluriel, masculinféminin)
                }
```

Un autre exemple de la variation du nombre de formes par paradigme selon la vue linguistique est la frontière entre suffixation et flexion. En suédois, on peut considérer la suffixation du déterminant défini comme faisant partie de la flexion. Les éléments ajoutés

sont *–en* au singulier et *–na* au pluriel, comme c'est le cas pour *administrationen* et *administrationerna*.

SV Pf (akademi, nom utrum) = {
(akademi, singulier indéfini)
(akademin, singulier défini)
(akademier, pluriel indéfini)
(akademierna, pluriel défini)
}

Dans certaines langues, le diminutif est formé par suffixation. La question se pose de l'inclure ou non dans le paradigme flexionnel. La marque du diminutif en néerlandais est le suffixe *je*, dont la suffixation peut demander l'ajout d'autre lettres de liaison ou orthographique.

NL	*huis + je*	= *huisje*	(petite maison)
NL	*boom + je*	= *boompje*	(petit arbre)
NL	*man + je*	= *mannetje*	(petit bonhomme)

L'inclusion ou non dans le paradigme flexionnel dépend de l'usage qu'on compte faire du lemme, donc de l'application. Cela dépend aussi de l'usage dans la langue : en néerlandais par exemple le diminutif est souvent utilisé pour exprimer une relation affective à l'objet en question. L'objet n'est pas pour autant vraiment plus petit : un *huisje* peut être une maison de proportions tout à fait normales dans son contexte. Un *mannetje* est en général utilisé avec dédain sans que l'homme en question soit pour autant petit. Un *boompje* peut aussi bien être un jeune qu'un petit arbre.

Une question similaire se pose pour les formes féminines des métiers et des titres. Est-ce que les formes de *princesse* et de *boulangère* se rangent dans les paradigmes respectifs de *prince* et de *boulanger* ou est-ce qu'ils on droit à leur propre entrée ? L'approche lexicologique consiste à les ranger avec leur forme masculine. Morphalou invoque sur sa page de présentation[115] « plusieurs raisons linguistiques […] à ne pas considérer le genre comme un trait flexionnel des noms communs en français (notamment le caractère non paradigmatique de la variation, la différence du sens dénotationnel et des collocations potentiellement différentes). » Contrairement aux choix des lemmes dans le TLF, « les formes féminines des noms communs concernés ne sont plus codées en tant que formes fléchies du lemme masculin, mais en tant que lemmes à part entière. » Les trois raisons invoquées sont valables mais pas déterminantes. *Primo*, qu'il existe une certaine irrégularité dans la formation du féminin plaide en effet en défaveur de leur inclusion dans les paradigmes flexionnels. Néanmoins elle peut être très régulière dans d'autres langues, comme en allemand par la suffixation de *–in* par exemple. *Secundo*, il nous semble que le sens dénotationnel entre *fourmi* et *fourmis* n'est pas le même non plus, pourtant cela ne sert pas d'argument pour ne pas inclure le pluriel dans un paradigme flexionnel. A cause de la différence de sens entre les différentes formes d'un paradigme, les collocations sont potentiellement différentes pour toutes les formes du même paradigme flexionnel, et non seulement pour une forme féminine. Cela pourrait se mesurer en comparant formes lemmatisées des collocations entre toutes les formes d'une sélection de paradigmes flexionnels. Objectivement, l'inclusion des formes féminines dans le paradigme aurait comme effet d'augmenter le nombre de mots-formes et de paradigmes flexionnels ainsi que de diminuer le nombre de lemmes. Les paradigmes flexionnels ne sont pas définis par la langue mais par le linguiste ou le lexicologue en raison des critères qu'il applique. Les paradigmes de flexion sont une modélisation de la langue que

[115] http://www.cnrtl.fr/lexiques/morphalou/LMF-Morphalou.php, consulté le 26 juin 2009.

son concepteur a formalisée en fonction des besoins exprimés. Si le concepteur veut garder son lexique générique, il doit prévoir un mécanisme qui permet de basculer d'un modèle à l'autre.

Le nombre de mots-formes dépend finalement du standard grammatical adopté. En néerlandais, selon la région, un système de deux ou de trois genres est d'usage pour les noms. Soit le neutre et le non neutre, soit le neutre, le masculin et le féminin. La différentiation entre masculin et féminin n'influe pas sur l'utilisation des articles comme on pourrait le penser, mais pour sur la référence pronominale. La grammaire de référence du néerlandais, [ANS, 2005], décrit la référence par le pronom personnel masculin à des non personnes (comme la *poêle*), qu'elles soient de genre masculin, féminin ou neutre : « en principe la seule forme pleine qui peut être utilisée dans tous les cas pour référer à des non personnes » (article 5·2·5·2·1 b). L'exemple suivant est repris tel quel de la même grammaire et exprime par la qualification « douteux » (et non pas incorrect) la référence qui fait l'accord avec le genre. Un peu plus loin, elle admet toutefois que « dans la pratique » elle existe bien.

NL *(Waar heb ik die pan ook weer gelaten?) O ja,* zij *staat op de vensterbank.* <<*twijfelachtig*>>
(Où est-ce que j'avais laissé cette poêle ?) Ah oui, *elle* est sur le rebord de la fenêtre. <<douteux>>

En effet, la norme prescrit l'utilisation du pronom masculin pour un référent de genre féminin. Si on applique cette règle à la lettre, elle admet l'exemple suivant, souvent fustigé comme un exemple de dégradation des normes linguistiques aux Pays-Bas.

NL *De koe,* hij *geeft melk.* (La vache, *il* donne du lait)

La réalité est plus nuancée : il existe des différences régionales. Dans certaines régions la reprise pronominale se fait exclusivement par le pronom personnel *hij*, tandis que dans d'autres, on adapte le pronom au genre du référent en utilisant le masculin *hij* ou le féminin *zij*/*ze*. Quelle langue faut-il décrire, la langue normée ou la réalité ? Et comment décrire la réalité dans un seul modèle si elle est multiple ?

Pour savoir quelles connaissances inclure, on applique souvent l'adage « Qui peut le plus peut le moins », qui va souvent à l'encontre de « *Less is beautiful* », qui est tout aussi appréciable dans un contexte où la masse de données est considérable. Pour le néerlandais, cela revient à décrire les genres dans les connaissances en appliquant le système des trois genres, car il peut, par règle, être réduit à deux, l'inverse n'étant pas vrai. La situation est d'autant plus complexe qu'il peut exister des appréciations régionales sur le genre des mots, alors que le genre de chaque mot est également indiqué dans la liste qui fixe l'orthographe officielle des mots du néerlandais [Woordenlijst, 2005]. Certains mots ont deux genres sans que leur paradigme de flexion ni leur valeur sémantique ne changent. Si c'était le cas, les mots auraient deux entrées différentes dans la liste (comme par exemple le mot néerlandais *diamant* qui est neutre et invariable en nombre s'il désigne la matière, mais qui est masculin et variable en nombre s'il désigne un morceau de la matière).

A cause de la relativité du paradigme de flexion, on peut considérer que l'ensemble des paradigmes déployés pour l'encodage d'un lexique font entièrement partie de ce lexique. En outre, à moins de standardiser les paradigmes pour chaque langue, les lexiques factorisés ne sont pas échangables. La question se pose même pour des lexiques non factorisés, car la fusion de deux lexiques peut mener à des incohérences au niveau des paradigmes de flexion inhérentes aux lexiques.

9.2 Une aide expérimentale à l'enrichissement semi-automatique

Il arrive à un certain moment dans la phase d'acquisition qu'une partie du lexique est en place, mais qu'on peine à changer d'échelle. Supposons que le lexique contienne les mots-formes correspondants aux 30 000 lemmes les plus fréquents, cela est sans doute largement suffisant pour couvrir de façon honorable la plupart des textes tout venants. Néanmoins, pour un traitement plus complet, la taille du lexique devrait doubler, voire tripler, tandis que les ressources lexicales originales sont épuisées. Dans ce cas, nous proposons une aide à l'enrichissement semi-automatique pour faciliter l'augmentation de la taille du lexique de base à partir de corpus. L'outil que nous avons mis en place peut également être utilisé pour la mise à jour.

Les chercheurs qui ont exploré la construction (semi-)automatique de lexique ont en commun d'introduire leurs articles avec les mêmes constats et besoins. [Zanchetta et Baroni, 2005] les exprime ainsi :

> "Lexicons and morphological analysers are at the core of many NLP applications, such as lemmatisation, POS tagging and morphology generation. Unfortunately, since the creation of a lexicon tends to be a long and labour intensive task (especially for highly inflectional languages), to date, there are no freely available lexicons for the Italian language."

La constitution d'un lexique est considérée par les informaticiens comme une tâche laborieuse, par les linguistes comme une tâche répétitive, et par les employeurs comme (trop) longue et donc coûteuse. Les motivations pour essayer d'automatiser cette tâche sont donc multiples. Certains chercheurs le font à des fins d'étude linguistique, comme [Goldsmith, 2001] qui essaie d'acquérir à partir d'un corpus et de façon non supervisée des modèles d'analyse morphologique qui sont linguistiquement motivés.

9.2.1 Approche générale

Depuis quelques années, différents travaux sont apparus concernant la constitution de lexique de façon (semi-)automatique, et la disponibilité de corpus plus grands facilitant des calculs probabilistes y est sans doute pour beaucoup. Toutes les approches de ces dernières années partent d'une ressource existante. Les paradigmes flexionnels étant primordiaux dans un bon nombre de ces travaux, [Zeman, 2008] montre comme les créer de façon non supervisée.

Dans la lignée de [Clément et al., 2004] qui a donné naissance au lexique Lefff, [Sagot, 2005] et [Sagot, 2007] affinent la méthode pour construire des lexiques morphologiques à large couverture en slovaque et en polonais. A partir d'une description morphologique de la langue sous forme de classes de flexion, tous les lemmes possibles sont générés pour les tokens du corpus absents du lexique. Les lemmes résultants sont ensuite classés selon un calcul de probabilité d'être corrects prenant en compte la diversité des mots-formes du même lemme dans le corpus et leur fréquence. Le calcul du lemme et la génération des mots-formes sont basés sur les classes de flexion. [Adolphs, 2008] applique la même méthode en allemand avec une variation sur le calcul probabiliste.

[Oliver et Tadic, 2004] présente la méthode suivante. A partir du lexique croate HML d'environ 25 000 lemmes correspondant à 1,5 millions de mots-formes, les auteurs déduisent de façon automatique des règles morphologiques. Les règles sont regroupées en paradigmes, et les paradigmes sont sélectionnés selon leur productivité dans le lexique. Pour tous les mots

du corpus qui n'apparaissent pas dans le lexique, tous les *stems* et terminaisons possibles sont calculées, avec quelques restrictions sur la taille du stem et de la terminaison. Les mot-formes sont regroupés par stems, et les paradigmes les plus complets sont considérés comme les meilleurs. L'évaluation est faite sur une liste de mots réguliers connus. Toujours sur le croate, [Šnajder et al., 2008] utilisent en entrée des classes de flexion et/ou des classes dérivationnelles dans un formalisme fonctionnel. Ils incluent les fréquences des tokens du corpus à analyser dans les calculs et ajoutent des heuristiques dépendantes de la langue pour améliorer les résultats.

En italien, [Zanchetta et Baroni, 2005] partent d'un corpus annoté automatiquement et appliquent différentes stratégies d'identification des lemmes selon la catégorie grammaticale à partir d'heuristiques linguistiques. Les mots-formes sont ensuite générés avec un fléchisseur.

[Nakov et al., 2003] travaillent uniquement sur les noms en allemand. A partir d'un corpus annoté, un lexique de taille réduite et des classes de flexion, ils induisent des règles d'association pondérées entre les terminaisons et les classes de flexion.

[Carlos et al., 2009] travaillent sur le hindi, et utilisent un stemmer basé sur des règles morphologiques et un étiqueteur de catégories grammaticales pour améliorer les performances.

La particularité de notre méthode[116] est qu'elle ne se base à aucun moment sur les paradigmes de flexion et n'essaie pas d'en créer. La principale source de connaissances est un lexique extensionnel précédemment constitué, à partir duquel nous créons des probabilités terminaison-description. Notre seconde source est le corpus. Grâce au web, il est aujourd'hui possible de collecter de très grands corpus. L'un d'entre eux qui apparaît dans de nombreux d'études et que nous exploitons ici est Wikipedia, qui a l'avantage d'être librement disponible sous le format HTML[117]. Nous tirons profit de la disponibilité de ce type de très grand corpus pour émettre des hypothèses sur les mots à ajouter dans le lexique, en nous basant sur les connaissances du lexique. Notre but est bien défini : fournir une aide à la personne qui code des entrées dans le lexique une fois qu'un lexique minimal est disponible. Nous ne cherchons pas à constituer un lexique complet à partir d'aucune ressource lexicale comme c'est le cas de la plupart des approches référencées. Notre but est partagé par [Loupy et al., 2009] qui cherche également à optimiser l'enrichissement lexical. Les auteurs proposent un lemme et un paradigme de flexion à partir des possibles terminaisons du token. Les probabilités sont calculées à partir du lien entre les terminaisons et les paradigmes correspondants selon les entrées du lexique. Cela est possible car le lexique de [Loupy et al., 2009] est construit en intension, c'est-à-dire comme un ensemble de couples <lemmes-descriptions>. N'ayant pas à disposition ce genre d'information, nous voulons éviter d'injecter des connaissances propres à chaque langue, ou être obligé de construire des paradigmes de flexion manuellement. Nous faisons donc le lien entre la terminaison, que nous ne calculons pas à partir du lemme, et la description.

9.2.2 Construction des connaissances à partir du lexique

Nous partons du principe que pour toute langue flexionnelle, l'essentiel du lexique est régi par un nombre de règles morphologiques restreint, ce qui est d'autant plus vrai pour les classes ouvertes et les néologismes. Notre lexique de base, aussi restreint soit-il, contient soit du

[116] Nos expériences ont été effectuées milieu 2007, une partie des travaux cités dans cet état de l'art n'étaient donc pas encore publiés.
[117] Disponibles sur : http://static.wikipedia.org/.

vocabulaire courant (mots grammaticaux et mots courants), soit des exceptions qui sont bien documentées dans les grammaires. Si elles méritent notre attention au tout début de la phase d'acquisition, l'augmentation du lexique porte surtout sur ce grand ensemble de mots qui ont des formations très régulières. L'idée générale est d'utiliser les informations lexicales existantes pour classifier des mots simples qui sont inconnus du lexique et de proposer un lemme et une catégorie grammaticale. L'expert vérifie ensuite la validité et ajoute le paradigme complet au lexique en utilisant le fléchisseur. L'augmentation du volume lexical accroît les performances de l'outil car les hypothèses émises sont plus précises. Le processus d'utilisation est illustré en figure 74.

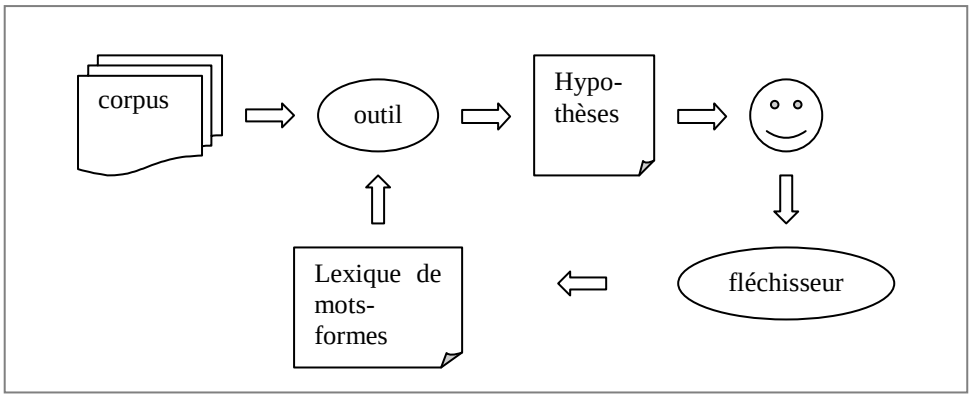


Figure 74 : Processus d'enrichissement lexical

La figure 75 montre un bref extrait d'un lexique morphosyntaxique à Sinequa. Comme nous l'avons décrit précédemment, une entrée lexicale est constituée d'un mot-forme, de son lemme sous forme abrégée (calculée à partir du mot-forme) et de la description morphosyntaxique. Le lemme et la description peuvent être multiples en cas d'ambiguïté, comme c'est le cas pour *manie* et *manient* dans l'exemple donné. Sur la même ligne, les ambiguïtés peuvent se porter sur plusieurs lemmes (*manie* – *manier*) ou seulement sur la description (*manient*).

```
%manie : () NOM.F.S : (r) VER.IM.TU : (r) VER.PI.IL : (r) VER.PI.JE : (r) VER.SUP.IL : (r) VER.SUP.JE
%maniement : () NOM.M.S
%maniements : (-1) NOM.M.P
%manient : (-2r) VER.PI.ILS : (-2r) VER.SUP.ILS
%manier : () VER.IN
```

Figure 75 : Extrait d'un lexique morphosyntaxique

Dans ces exemples, la différence de lemme coïncide avec une catégorie grammaticale différente. Ce n'est pas toujours le cas, comme par exemple pour *manger* (figure 76) ou *avoir* : le lemme est effectivement le même pour le nom et pour le verbe.

```
%mangeottés : (-2er) PPS.M.P
%manger : () NOM.M.S : () VER.IN.VDAT
%mangera : (-1) VER.FI.IL.VDAT
```

Figure 76 : Extrait d'un lexique morphosyntaxique

Que le lemme ne soit pas un identifiant unique, mais un des mots-formes du paradigme choisis selon la convention habituelle pose problème pour les rares cas où le mot-forme est le

même au singulier mais différent au pluriel (comme à c*iglio*/*cigli* et *ciglio*/*ciglia* en italien, voir 5.6), ainsi que dans les cas ou les *singularia tantum* coexistent avec d'autres mots-formes identiques au singulier (comme *sel*, l'aliment, et *sel/sels* le composé chimique). Dans le premier cas, le paradigme construit risque d'être faux car il en rassemble en réalité deux, et dans le second cas, les entrées des singularia seront ignorées au profit des autres mots-formes. Toutes nos expériences ont été conduites sur le lexique allemand.

A partir du lexique morphosyntaxique de base, nous avons construit la liste des terminaisons de chaque mot en comptant leur nombre d'occurrences en combinaison avec les descriptions. Nous considérons comme terminaisons d'un mot toutes les suites de lettres à partir de la 2^{e} lettre du mot jusqu'à la fin du mot. Les terminaisons pour le mot *abcd* sont donc : *d*, *cd*, *bcd*.

Les prétraitements suivants ont été exécutés sur le lexique. L'ordre des étiquettes étant libre, nous l'avons normalisé (par exemple NOM.F.S.G > NOM.G.F.S). Pour rester dans les descriptions des paradigmes flexionnels, nous avons enlevé les étiquettes qui n'en relevaient pas, comme par exemple NPR (nom propre) et MOIS (mois). Nous avons également enlevé les commentaires. Finalement, nous avons exclu les entrées dont le mot-forme comporte un tiret ou un espace.

La liste, construite sur 392 248 mots-formes et leurs descriptions, contient 1 754 832 combinaisons uniques de terminaison-descriptions. Le nombre de terminaisons uniques est de 870 096. La liste se caractérise par un petit nombre de terminaisons courtes très fréquentes et un très grand nombre de terminaisons longues. Le début et la fin de cette liste sont donnés en tableau 22.

Terminaison-description	Fréquence
n-NOM.D.F.P	13731
n-NOM.N.F.P	13609
n-NOM.A.F.P	13601
n-NOM.G.F.P	13600
en-NOM.D.F.P	13447
en-NOM.N.F.P	13327
en-NOM.A.F.P	13319
en-NOM.G.F.P	13318
s-NOM.G.M.S	11174
t-VER.SUP.TU	9612
st-VER.SUP.TU	9612
e-VER.SUP.IL	9598
e-VER.SUP.JE	9590
…	…
mokratisieren-VER.SUP.ILS	1
etuckerten-VER.PPS.EN	1
chäftiger-ADJ.ER	1
ßtischstes-ADJ.SUPER.ES	1
msiedlungen-NOM.G.F.P	1
ammenschlüsse-NOM.A.M.P	1
hndung-NOM.G.F.S	1
tent-NOM.D.X.S	1

Tableau 22 : Listes de terminaison-descriptions et leur fréquence

La liste est inexploitable telle quelle. Le nombre de terminaison-descriptions est extrêmement grand, les terminaison-descriptions les plus fréquentes sont trop courtes et trop ambiguës pour être signifiantes et la grande majorité des terminaison-descriptions n'apparaissent qu'un nombre très limité de fois. Le diagramme de Pareto de la figure 77 illustre la proportion très inégale de combinaison-terminaisons selon leur fréquence : 94,68 % de ces combinaisons apparaissent jusqu'à 5 fois.

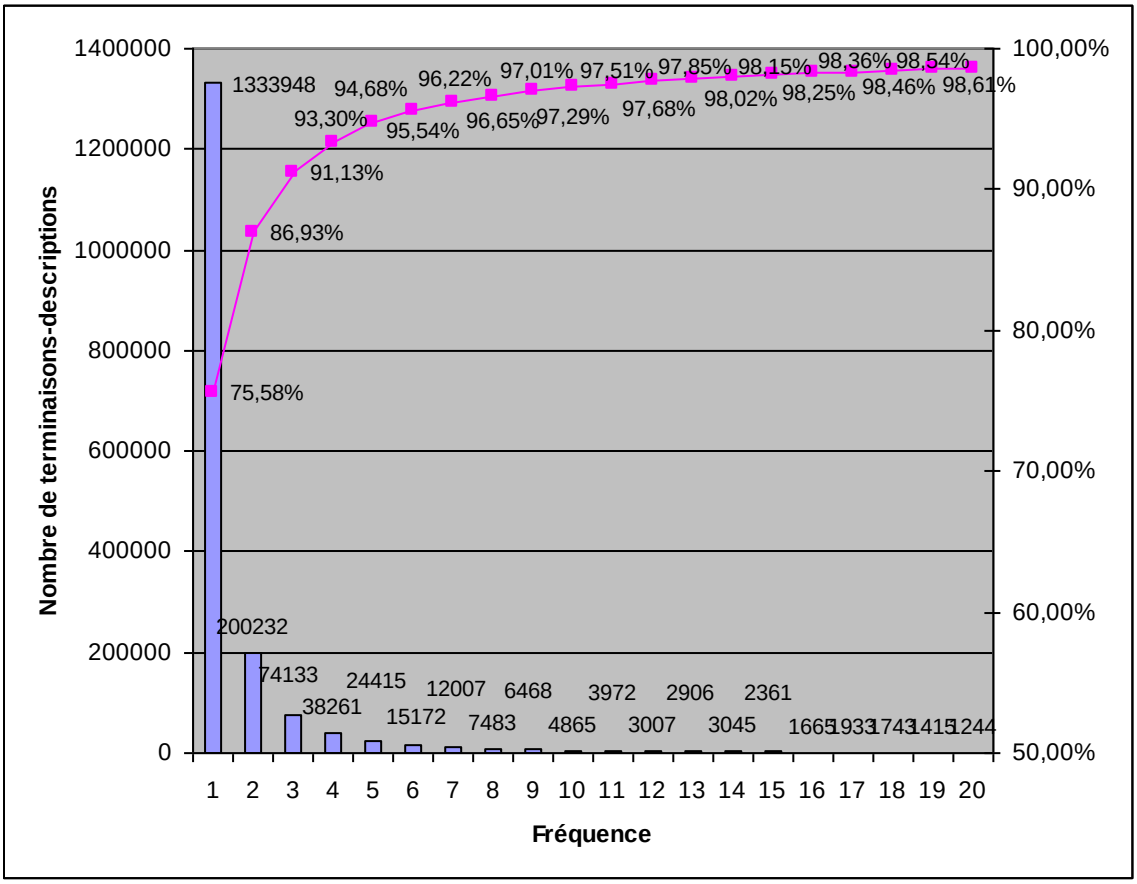


Figure 77 : Diagramme de Pareto pour les terminaisons-descriptions selon la fréquence

Pour réduire le bruit causé par notre calcul très brutal des terminaisons, nous avons enlevé de la liste toutes les combinaisons qui apparaissent 10 fois ou moins. Ce seuil nous paraît optimal par rapport à nos observations. [Nakov et al., 2003] mettent également la limite basse des occurrences des terminaisons à 10. Ce filtrage fait tomber le nombre de combinaisons de terminaisons-descriptions de 1 764 832 à 47 854, ramenant le nombre de terminaisons uniques de 870 096 à 22 081. Les terminaisons concernées sont celles qui couvrent quasiment le mot complet.

Le nombre de terminaisons uniques est toujours assez grand, et beaucoup d'entre elles se recouvrent, la fin étant identique, comme par exemple les terminaisons *ichkeiten* et *chkeiten*. Il est donc possible de réduire leur nombre en remplaçant la terminaison plus longue par la terminaison plus courte sous condition qu'elles aient le même pouvoir informationnel.

Nous calculons le pouvoir informationnel de chaque terminaison en calculant l'entropie de chaque description sachant la terminaison. Le score d'entropie permet d'ordonner la liste des terminaisons en fonction de leur pouvoir informationnel.

Le calcul de l'entropie se base sur la probabilité d'apparition d'une description sachant la terminaison. Nous définissons l'entropie *H* de la terminaison *t* de façon classique comme suit :

$$H(t) = -\sum_i p(d_i|t)\log(p(d_i|t))$$

Nous calculons la probabilité *p* de *i* en divisant le nombre d'apparitions de *i* par la somme totale des apparitions de tous les *i* :

$$p(d_i|t) = \frac{n(d_i,t)}{\sum_j n(d_j,t)}$$

Plus le score d'entropie s'éloigne de 0, plus le pouvoir informationnel est faible. Le tri des terminaisons selon leur score d'entropie permet de déterminer les terminaisons les plus intéressantes à partir du nombre de fois où elles apparaissent. Les terminaisons avec le meilleur score d'entropie (0) sont toutes des terminaisons non ambiguës : elles n'ont qu'une seule description. Si un mot se termine par l'une d'entre elles, la probabilité que ce soit la bonne description est maximale. Les terminaisons les plus longues ont un score généralement inférieur aux terminaisons courtes.

Si un mot se termine par une terminaison à l'entropie 0, comme c'est le cas de 14 388 d'entre elles, on peut considérer que la description est la bonne. Un pouvoir informationnel plus faible ne signifie pas pour autant que la terminaison est moins importante.

Le score d'entropie nous donne un moyen de structurer et d'ordonner cette liste. Son évolution est comprise entre 0 et 3,400, ce que nous avons visualisé dans la figure 78. 14 388 terminaisons n'ont qu'une seule description, ce qui se visualise par un premier plateau à entropie zéro. 2 533 terminaisons (de 14 389 à 16 922) ont deux descriptions. Leur score monte rapidement pour se stabiliser à 0,69, ce qui se visualise par une première pente et un deuxième plateau. Ensuite le nombre de terminaisons diverge, mais est toujours supérieur à deux. Pentes et plateaux s'alternent pour se terminer dans une longue pente.

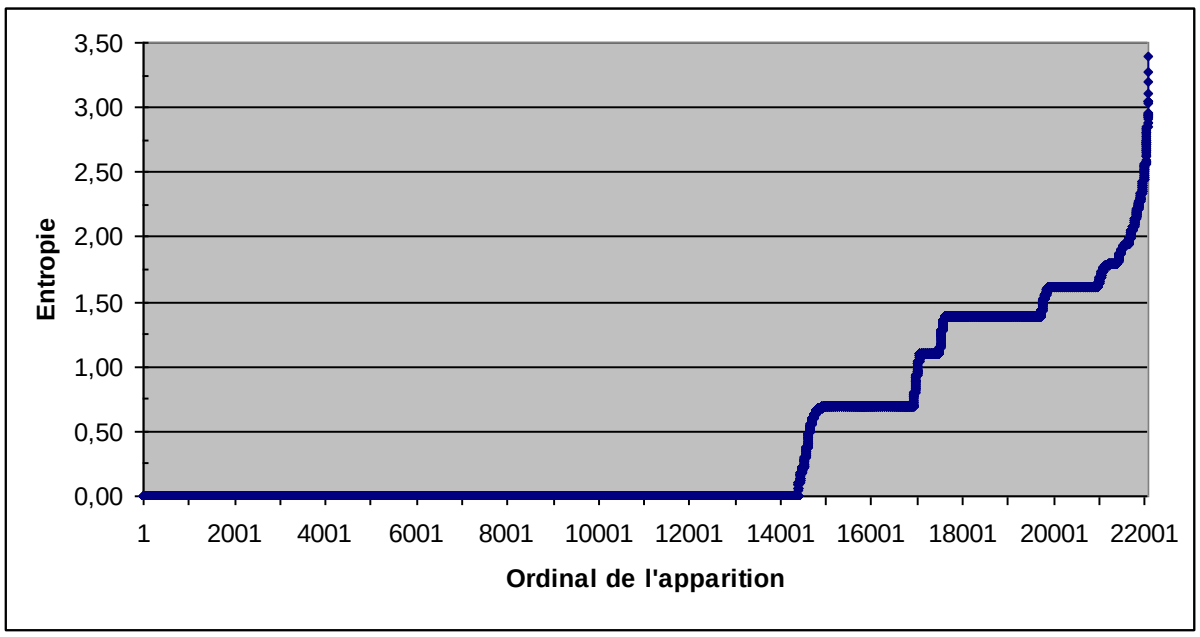


Figure 78 : Evolution de l'entropie

Pendant la première pente, il existe un fort déséquilibre entre les fréquences des deux descriptions comme par exemple dans le résultat suivant à la position 14 407 :

lende : 0,107 [VER.IM.TU-**12**][VER.PPR.E-**523**]

Les différences s'effacent progressivement et arrivent à un équilibre presque parfait autour de 15 000 comme dans l'exemple suivant qui se trouve à la position 15 045 :

hischer : 0,692 [ADJ.COMP.0-**40**][ADJ.ER-**43**]

Juste après la différence s'efface et les fréquences s'égalisent. Un phénomène similaire s'observe à la pente suivante autour du rang 17 000 qui est marqué par l'ajout d'une troisième description. On constate d'abord que l'entropie d'une des trois descriptions est nettement plus basse que les deux autres, alors que cette différence se lisse et s'efface peu après.

Comme nous avons remarqué qu'un grand nombre de terminaisons sont des sous-chaînes de terminaisons plus longues, nous avons rationalisé la liste par élimination des super-chaînes à partir de la fin du mot et dans la limite du même score d'entropie. Ainsi *lnder* représente 35 autres terminaisons qui ont toutes le même score d'entropie (0) comme *telnder*, *ppelnder*, *tzelnder*, *ckelnder*, *ndelnder*, *ügelnder*, *sselnder*, *ickelnder*, *nzelnder*, *mmelnder*, etc. Au lieu des 22 082 terminaisons, la liste a ainsi été réduite à moins d'un tiers de sa taille originale : 6 962.

Nous avons réalisé cette rationalisation par la construction d'une forêt d'arbres de terminaisons, dont nous n'avons gardé que les racines. Tous les nœuds d'un même arbre ont le même score d'entropie. Le père du nœud courant est une sous-chaîne de ce nœud à partir de la fin de la chaîne. La racine ainsi obtenue est la sous-chaîne la plus courte commune à tous les nœuds de l'arbre, toujours à partir de la fin de la chaîne. On coupe ensuite les descendants pour ne garder que la racine et sa probabilité, qui représente alors l'ensemble des sous-chaînes élaguées.

Nous illustrons l'arbre obtenu pour la racine *keiten* dans la figure 79. Tous les éléments de cet arbre ont le même score d'entropie. Quand deux racines ont le même nombre de lettres, les deux constituent des arbres différents, comme par exemple *keiten* et *heiten* (1,386). La racine *eiten* ne fait pas partie du même ensemble, car elle a un score d'entropie plus élevé (1,689). La racine *eiten* a d'ailleurs un score d'entropie unique et n'a donc aucun descendant.

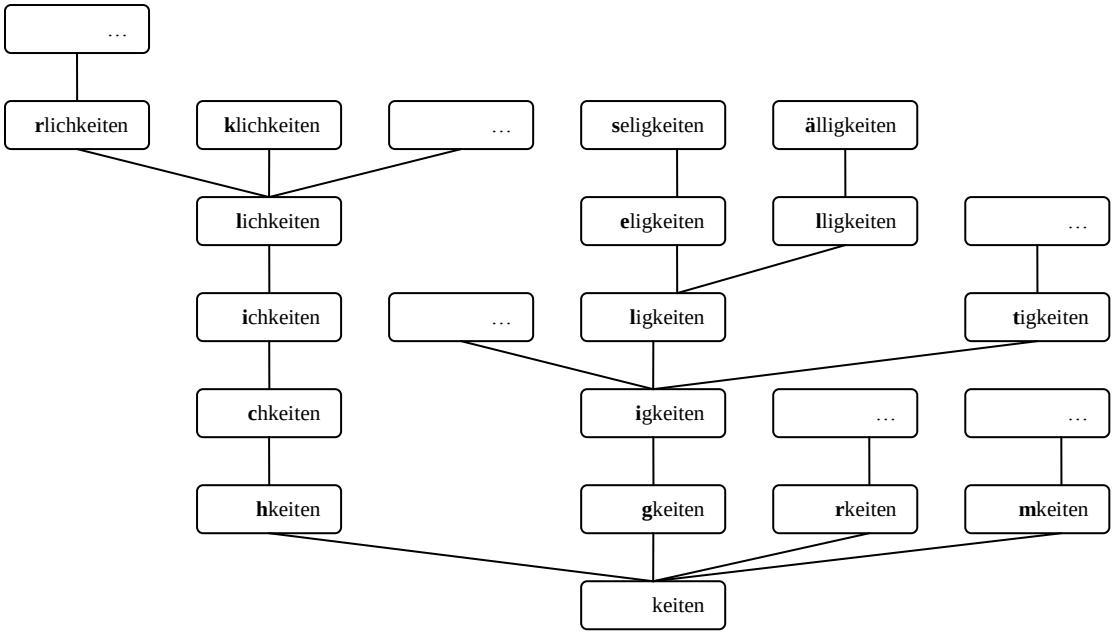


Figure 79 : Arbre de terminaisons, racine *keiten*, p=1,38629436111989

L'énorme réduction du nombre de terminaisons se fait grâce au fait que trois quarts (16 922 des 22 082) des terminaisons originales ont le même score d'entropie : elles ont un score de 0 ou de 0,693 comme nous avons vu ci-dessus en figure 78. Une fois passée ce dernier score, le score d'entropie double rapidement et est beaucoup moins homogène. Nous aurions pu introduire des tranches pour éviter des séparations artificielles à cause d'un écart très faible de quelques points, mais nous ne nous sommes pas engagé dans cette voie.

Nous avons ajouté un filtre supplémentaire qui peut être paramétré en fonction de la situation d'acquisition, selon qu'on souhaite se concentrer sur des phénomènes plus ou moins courants. Ce filtre constitue un élagage de plus qui sélectionne les terminaisons dont la somme des occurrences des descriptions dépasse un certain seuil. Si le seuil est à 0, *fischer* est une racine prise en compte avec les informations suivantes : fischer : 0,667 [ADJ.COMP.0-**12**][ADJ.ER-**19**]. Avec un seuil à 50, cette racine sera ignorée, car 12+19 < 50. Nous apellerons ce seuil le *seuil cumulatif des terminaisons*.

L'impact sur le nombre de terminaisons de ce seuil est donné dans le tableau 23 et est visualisé dans la figure 80. On y voit que le pourcentage de terminaisons avec une probabilité de moins de 0,70 se normalise fortement avec un seuil entre 200 et 800 : il varie entre 25,15 % et 28,05 %. Au-delà de 800, le pourcentage affaiblit, et l'intérêt également car le nombre total de terminaisons devient moins intéressant. Comme les pourcentages sont plus hauts avec un seuil en dessous de 200, nous le conseillons comme seuil minimal. Pour la suite de nos expériences, nous l'avons mis à 500.

Seuil	Nb terminaisons (A)	Nb terminaisons p < 0,70 (B)	B/A
0	6962	4255	61,12%
25	5278	2571	48,71%
50	4016	1524	37,95%
75	3042	1054	34,65%
100	2387	755	31,63%
125	2027	616	30,39%
150	1721	508	29,52%
175	1549	448	28,92%
200	1383	372	26,90%
225	1262	338	26,78%
250	1155	313	27,10%
275	1065	286	26,85%
300	1001	269	26,87%
325	934	253	27,09%
350	863	229	26,54%
375	806	211	26,18%
400	749	200	26,70%
425	716	192	26,82%
450	685	183	26,72%
475	656	181	27,59%
500	631	177	28,05%

Tableau 23 : Rapport Seuil- Nombre de terminaisons

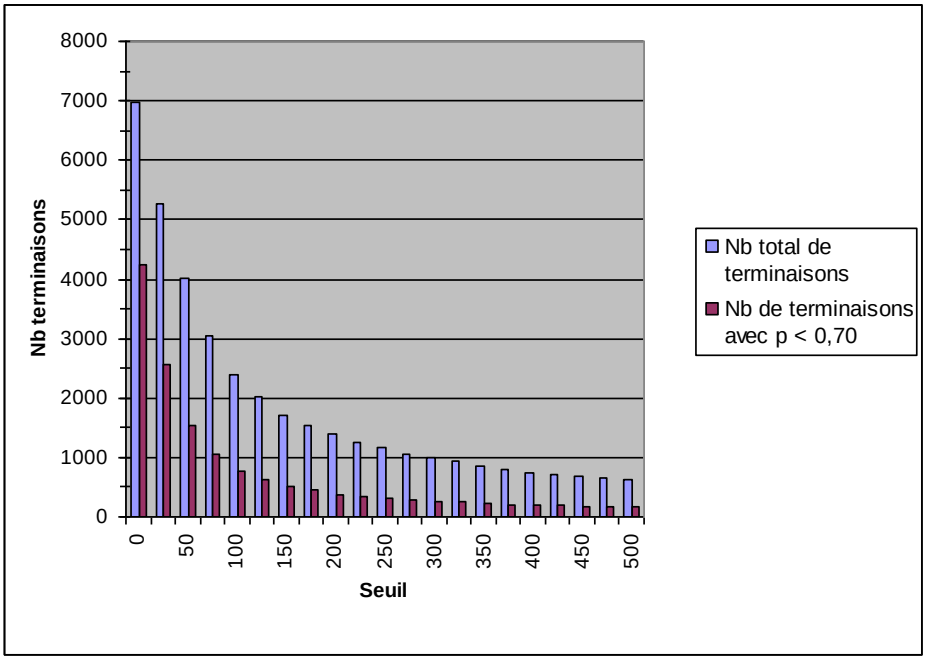


Figure 80 : Rapport Seuil - Nombre de terminaisons

Un extrait des terminaisons résultant après élagage avec un seuil à 500 est donné dans le tableau 24. Il mentionne quelques entrées avec un score d'entropie à 0 ainsi que le score le plus haut (3,400). Les descriptions associées aux terminaisons sont assorties du nombre de

fois où elles apparaissent dans le lexique. Dans ce tableau, nous avons choisi de mettre l'exemple de *render*, *ender*, *nder* et *der*, qui sont trois terminaisons qui auraient fait partie du même arbre si leurs scores avaient été les mêmes. Or, ce n'est pas le cas, et le score d'entropie ne cesse de monter. Le nombre d'ambiguïtés monte également. Nous avions vu dans le tableau 22, page 213, que la terminaison *n* est la plus courante. Avec l'entropie, cela s'est traduit par un classement au plus bas de la liste.

Termi-naison	Score d'entropie	Descriptions associées à la terminaison avec leur nombre d'occurrences
lendes	0	[VER.PPR.ES-523]
tester	0	[ADJ.SUPER.ER-507]
herem	0	[ADJ.COMP.EM-1317]
hsten	0	[ADJ.SUPER.EN-1088]
iges	0	[ADJ.ES-746]
hem	0,046	[ADJ.EM-1622][NOM.D.X.S-13]
render	0,060	[ADJ.ER-13][VER.PPR.ER-1171]
hend	0,138	[ADJ.0-29][VER.PPR.0-907]
end	0,179	[ADJ.0-194][NOM.N.M.S-14][VER.IM.TU-33][VER.PI.JE-33][VER.PPR.0-7804]
ender	0,345	[ADJ.COMP.0-138][ADJ.ER-193][NOM.A.M.P-18][NOM.A.M.S-18] [NOM.D.M.S-18][NOM.G.F.P-29][NOM.G.M.P-45][NOM.N.M.P-18] [NOM.N.M.S-45][VER.PPR.ER-7804]
nder	0,589	[ADJ.COMP.0-150][ADJ.ER-206][NOM.A.M.P-89][NOM.A.M.S-87] [NOM.A.X.P-12][NOM.D.M.S-87][NOM.G.F.P-32][NOM.G.M.P-119] [NOM.G.X.P-12][NOM.N.M.P-89][NOM.N.M.S-117][NOM.N.X.P-12] [VER.IM.TU-29][VER.PI.JE-29][VER.PPR.ER-8989]
der	0,926	[ADJ.COMP.0-184][ADJ.ER-264][NOM.A.M.P-158][NOM.A.M.S-158] [NOM.A.X.P-43][NOM.A.X.S-24][NOM.D.M.S-158][NOM.D.X.S-24] [NOM.G.F.P-42][NOM.G.F.S-11][NOM.G.M.P-196][NOM.G.X.P-43] [NOM.N.M.P-158][NOM.N.M.S-196][NOM.N.X.P-43][NOM.N.X.S-24] [VER.IM.TU-96][VER.PI.JE-97][VER.PPR.ER-8989]
ur	2,100	[NOM.A.F.S-149][NOM.A.M.S-148][NOM.A.X.S-18][NOM.D.F.S-149] [NOM.D.M.S-148][NOM.D.X.S-18][NOM.G.F.S-148][NOM.N.F.S-149] [NOM.N.M.S-153][NOM.N.X.S-18]
n	3,400	[ADJ.0-270][ADJ.COMP.EN-3891][ADJ.EN-5090][ADJ.SUPER.EN-4137] [NOM.A.F.P-13601][NOM.A.F.S-2052][NOM.A.M.P-2967][NOM.A.M.S-2645] [NOM.A.X.P-821][NOM.A.X.S-1090][NOM.D.F.P-13731][NOM.D.F.S-2319] [NOM.D.M.P-8691][NOM.D.M.S-2649][NOM.D.MX.P-47][NOM.D.X.P-2975] [NOM.D.X.S-1135][NOM.G.F.P-13600][NOM.G.F.S-2436][NOM.G.M.P-2966] [NOM.G.M.S-1996][NOM.G.X.P-822][NOM.G.X.S-45][NOM.N.F.P-13609] [NOM.N.F.S-2053][NOM.N.M.P-2965][NOM.N.M.S-671][NOM.N.X.P-821] [NOM.N.X.S-1092][VER.II.IL-28][VER.II.ILS-9133][VER.II.JE-28] [VER.II.NOUS-9133][VER.IM.TU-250][VER.IN-9029][VER.PI.IL-15] [VER.PI.ILS-8999][VER.PI.JE-292][VER.PI.NOUS-8999][VER.PPR.EN-8989] [VER.PPS.0-2391][VER.PPS.EN-9070][VER.SUI.ILS-9259][VER.SUI.NOUS-9259] [VER.SUP.ILS-9030][VER.SUP.NOUS-9030][VER.ZU-4760]

Tableau 24 : Extrait de la liste après élagage avec un seuil à 500

9.2.3 Construction des hypothèses sur le corpus

Pour construire les hypothèses de mots à ajouter, nous avons pris comme corpus la version HTML du Wikipedia allemand, datant du 1er juillet 2008. Wikipedia est structuré comme un ensemble d'articles, chaque article étant accompagné des commentaires de collaborateurs.

Nous avons soumis les tokens des documents de Wikipedia à l'analyseur et gardé ceux auxquels il n'a pas réussi à attribuer une catégorie grammaticale. Nous appelons ces tokens

des tokens *inconnus*. L'analyse prend en compte les entrées du lexique ainsi que les règles d'analyse morphosyntaxique. Le nombre s'élève à près de 5,8 millions de tokens inconnus. Un sixième de ces inconnus sont des noms de fichier et des adresses web que nous avons filtrés à l'aide de quelques expressions régulières ciblées. Les résultats de l'analyse sont donnés en tableau 25, avec la répartition des tokens inconnus uniques en tokens capitalisés, non capitalisés et filtrés.

Inconnus uniques	Inconnus uniques capitalisés	Inconnus uniques non capitalisés	Inconnus filtrés
5 795 002	3 670 999	1 173 119	950 884

Tableau 25 : Nombre d'inconnus lors de l'analyse du Wikipedia DE

Pour une autre langue que l'allemand nous aurions pu nous concentrer uniquement sur les tokens non capitalisés, mais les règles orthographiques allemandes prescrivent la capitalisation des noms et des nominalisations de verbe et d'adjectif. De ce fait, nous ne pouvons pas utiliser la capitalisation comme filtre pour les noms propres, qui sont omniprésents dans Wikipedia.

Nous avons associé une description à chaque token à partir de la liste des terminaisons obtenue avec un seuil à 500, préalablement ordonnée par le score d'entropie. 1 001 602 tokens non capitalisés et 2 664 041 tokens capitalisés ont ainsi été classifiés. 1 178 475 tokens tombent donc hors de l'espace des hypothèses.

La distribution des tokens classifiés est indiquée dans le tableau 26 selon la capitalisation et leur provenance d'articles ou de commentaires.

Tokens inconnus	Total	Commentaires	Articles
Non capitalisés	1 001 602	635 722	365 880
Capitalisés	2 664 041	939 855	1 724 186
Total	3 665 643	1 575 577	2 090 066

Tableau 26 : Répartition des tokens inconnus selon capitalisation et appartenance

Après avoir constaté que les commentaires apportent un bruit énorme, nous les avons écartés du corpus. Ils contiennent en effet beaucoup de fautes de frappe, ce qui génère énormément de mauvaises hypothèses. Cette version du Wikipedia allemand, nettoyé des commentaires avec un outil interne, contient un peu plus de 365 millions de tokens[118] (sans ponctuation), représentant un peu plus de 8,6 millions de tokens uniques (sensibles à la casse).

Nous avons ensuite regroupé les hypothèses par racine commune, ce qui donne un résultat comme dans le tableau 27, où on voit un petit extrait des hypothèses avec une fréquence de racine égale à deux.

[118] Utilisant tous les signes de ponctuation et d'espacement comme séparateurs, sauf le tiret entre deux tokens sans espacement.

Fréq. rac.	Racine	Termi-naison	Score d'entropie	Description
2	Rekrutierungsst	elle	2,453	[ADJ.E-81][NOM.A.F.S-64][NOM.A.M.P-14] [NOM.A.X.P-19][NOM.D.F.S-59][NOM.G.F.S-59] [NOM.G.X.P-19][NOM.N.F.S-64][NOM.N.M.P-14] [NOM.N.X.P-19][VER.IM.TU-95][VER.PI.JE-97] [VER.SUP.IL-97][VER.SUP.JE-97]
2	Rekrutierungsst	ellen	2,483	[ADJ.EN-81][NOM.A.F.P-68][NOM.D.F.P-68] [NOM.D.M.P-16][NOM.D.X.P-21][NOM.G.F.P-68] [NOM.N.F.P-68][VER.IN-97][VER.PI.ILS-97] [VER.PI.NOUS-97][VER.SUP.ILS-97] [VER.SUP.NOUS-97][VER.ZU-74]
2	Allosaur	ier	1,770	[ADJ.ER-17][NOM.A.M.P-82][NOM.A.M.S-133] [NOM.A.X.S-36][NOM.D.M.S-133][NOM.D.X.S-36] [NOM.G.M.P-82][NOM.N.M.P-82][NOM.N.M.S-134] [NOM.N.X.S-36][VER.IM.TU-842][VER.PI.JE-841]
2	Allosaur	us	1,913	[NOM.A.M.P-40][NOM.A.M.S-829] [NOM.A.X.P-30][NOM.A.X.S-30][NOM.D.M.P-40] [NOM.D.M.S-829][NOM.D.X.P-30][NOM.D.X.S-30] [NOM.G.M.P-40][NOM.G.M.S-863][NOM.G.X.P-30] [NOM.G.X.S-56][NOM.N.M.P-40][NOM.N.M.S-829] [NOM.N.X.P-30][NOM.N.X.S-30][VER.IM.TU-24] [VER.PI.JE-24]
2	Agrarwirts	chaft	1,386	[NOM.A.F.S-144][NOM.D.F.S-144][NOM.G.F.S-144] [NOM.N.F.S-144]
2	Agrarwirts	chaften	1,386	[NOM.A.F.P-142][NOM.D.F.P-142][NOM.G.F.P-142] [NOM.N.F.P-142]
2	Ahmadinescha	d	1,095	[ADJ.0-266][NOM.A.F.S-29][NOM.A.M.S-178] [NOM.A.X.S-184][NOM.D.F.S-29][NOM.D.M.S-178] [NOM.D.X.S-184][NOM.G.F.S-29][NOM.N.F.S-30] [NOM.N.M.S-209][NOM.N.X.S-185][VER.II.IL-146] [VER.II.JE-146][VER.IM.TU-280] [VER.IM.VOUS-30][VER.PI.ILS-29] [VER.PI.JE-283][VER.PI.NOUS-29] [VER.PI.VOUS-29][VER.PPR.0-8989]
2	Ahmadinescha	ds	1,753	[NOM.A.M.P-27][NOM.A.X.P-16][NOM.D.M.P-27] [NOM.D.X.P-16][NOM.G.M.P-27][NOM.G.M.S-178] [NOM.G.X.P-16][NOM.G.X.S-187][NOM.N.M.P-27] [NOM.N.X.P-16]

Tableau 27 : Extrait du résultat des hypothèses

Ces tokens inconnus doivent être interprétés par un expert. Le token *Rekrutierungsstelle* est un mot composé dont il manque un composant dans le lexique. Le token complet n'est pas à ajouter, mais le composant manquant, *Rekrutierung*, doit l'être. Certains mots sont peut-être trop spécifiques pour être ajoutés : est-ce que *Allosaurus* et *Allosaurier* ont leur place dans un lexique de large couverture ? Avec plus de 40 000 pages retrouvées sur Google si on les pose en requête, nous pensons que oui, même si cette méthode de jugement est très critiquable. *Agrarwirtschaft* pose le problème des préfixes : faut-il les traiter par règle ou par entrée lexicale ? Sachant qu'on peut écrire l'équivalent *Agrar-Wirtschaft*, une règle est sans doute la meilleure solution. Et même si nous n'avons pas l'intention d'ajouter des noms propres dans le lexique, il ne faut pas oublier que ceux-là se déclinent aussi et apparaissent à ce titre dans les hypothèses.

Cet extrait n'est pas représentatif de l'ensemble des résultats. Ce ne sont que quelques exemples choisis dans un ensemble où la qualité des hypothèses est variable. Sans pouvoir

donner de chiffres exacts à cause du grand nombre d'hypothèses, nous estimons que moins de 10 % des hypothèses concernent des mots à ajouter au lexique. Nous avons constaté que les résultats sont bien meilleurs sur les tokens non capitalisés, alors qu'en nombre de tokens uniques, ils sont bien moins nombreux comme on peut voir dans le tableau 28. Nous y donnons le nombre de racines selon le nombre de différentes terminaisons avec lesquelles elles apparaissent dans le corpus. Les exemples du tableau 27 font par exemple partie de l'ensemble « 2 », car les racines apparaissent dans le corpus avec deux terminaisons différentes.

	1	2	3	4	5	6	7	8	9	10	>10
Capit.	1 140 624	209 344	82 015	45 415	28 413	21 163	17 234	13 821	12 163	10 543	136 519
Non c.	176 086	51 849	32 774	20 507	15 571	9 057	6 222	4 809	3 629	2 921	40 874

Tableau 28 : Racines uniques des tokens inconnus (Wikipedia DE)

Pour les tokens capitalisés nous avons obtenu les meilleurs résultats pour les racines qui apparaissent avec 2 ou 3 différentes terminaisons. Pour les tokens non capitalisés, les meilleurs résultats sont dans la plage de 3 à 8, et les résultats sont plus pertinents. Au-delà de ces chiffres, la qualité se dégrade très vite.

Ces résultats semblent logiques. Les tokens non capitalisés sont surtout des verbes et des adjectifs : comme ceux-ci ont de nombreuses terminaisons, la plage est plus étendue. Les tokens capitalisés sont surtout des noms, dont les terminaisons sont moins nombreuses et moins variées. Pour les tokens capitalisés, la plage pertinente donc est plus courte. Le grand nombre de racines apparaissant qu'une seule fois est dû au grand nombre de noms propres qu'on retrouve dans Wikipedia. On trouve encore des hypothèses correctes dans les ensembles 7 et 8 qui sont des adjectifs de noms propres (d'où la capitalisation).

9.2.4 Filtrage par un seuil d'occurrences

Dans le souci de réduire le nombre d'entrées pour augmenter la pertinence, nous appliquons un filtre en utilisant un *seuil d'occurrences* des tokens. Cela enlève les tokens dont le nombre d'occurrences est en dessous du seuil. Nous présentons les conséquences quantitatives pour différents seuils dans le tableau 29 et le tableau 30, respectivement pour les tokens capitalisés et non capitalisés.

Seuil	1	2	3	4	5	6	7	8	9	10	>10
0	1 140 624	209 344	82 015	45 415	28 413	21 163	17 234	13 821	12 163	10 543	136 519
1	522 627	132 004	54 631	30 597	18 992	14 506	11 724	9 534	8 479	7 216	102 021
5	197 251	66 221	29 160	16 734	10 552	8 166	6 555	5 414	4 830	4 161	63 415
10	122 250	46 006	20 626	12 220	7 671	6 072	4 879	4 051	3 604	3 120	49 364
20	68 090	29 787	13 995	8 313	5 252	4 191	3 431	2 844	2 537	2 164	36 476
100	11 441	7 630	4 266	2 644	1 716	1 326	1 112	919	831	733	13 716
500	1 004	1 004	856	598	372	312	248	215	212	192	3 422
1000	346	455	412	290	203	171	132	106	124	118	1 769

Tableau 29 : Impact du seuil des occurrences sur le nombre de racines uniques par nombre de terminaisons.

Seuil	1	2	3	4	5	6	7	8	9	10	>10
0	176 086	51 849	32 774	20 507	15 571	9 057	6 222	4 809	3 629	2 921	40 874
1	60 847	25 590	19 540	12 835	10 348	5 724	3 986	2 987	2 192	1 759	24 954
5	15 501	7 959	7 961	5 644	5 249	2 813	1 969	1 378	1 068	848	11 562
10	7 616	4 158	4 659	3 366	3 459	1 889	1 335	928	730	546	7 594
20	3 244	2 002	2 393	1 785	2 093	1 201	861	563	448	355	4 817
100	334	273	347	270	389	277	196	108	111	113	1 372
500	21	22	22	19	21	29	17	9	14	20	287
1000	4	3	3	7	5	7	6	4	8	7	142

Tableau 30 : Impact du seuil des occurrences sur le nombre de racines uniques par nombre de terminaisons.

Des exemples donnés dans le tableau 27, seuls *Allosaurus* (89) et *Agrarwirtschaft* (335) survivraient à un seuil de plus de 10 occurrences, leurs pluriels n'apparaissant que deux et trois fois. *Rekrutierungsstelle* apparaît exactement 10 fois, ce qui montre que tout seuil est un choix arbitraire qui fait perdre de bons mots. Son pluriel apparaît quatre fois. En ce qui concerne *Ahmadineschad*, il est l'une des variantes des orthographes suivantes qu'on trouve toutes dans Wikipedia :

DE Ahmadinedschad (260), Ahmadinejad (23), Ahmadineschad (7), Ahmadinezhad (1)

Nous avons constaté que la pertinence augmente avec le seuil aussi bien pour les tokens non capitalisés que pour les capitalisés. Or, pour les tokens non capitalisés, il n'y pas besoin d'augmenter le seuil au-delà de 10, vu la bonne qualité déjà obtenue avec ce seuil. En plus, on risquerait de passer à côté de certains mots qui ont une moindre fréquence.

En ce qui concerne les tokens capitalisés, la pertinence augmente avec le seuil, mais les résultats ne sont pas très satisfaisants. Avec le seuil à 100 les noms communs restent toujours confondus dans la masse des noms propres. Avec un seuil à 1000, les résultats ne sont pas exceptionnels, même s'ils sont meilleurs, avec environ un mot sur 40 qui est un nom commun. Il est néanmoins intéressant d'étudier quels sont ces noms propres très fréquents que nous ne couvrons pas avec nos lexiques. Nous supposons que la variation de ce seuil peut donner de meilleurs résultats avec un lexique moins développé, mais cela reste à confirmer.

Comme l'effet de ce seuil est différent selon la capitalisation, qu'il peut servir à limiter la masse de données et qu'il est dépendant du corpus, nous le laissons à l'appréciation de l'utilisateur.

9.2.5 Evaluation

Le seul cadre d'évaluation qui nous semble valable pour ce type d'outil est le suivant. Comme c'est un outil d'enrichissement lexical destiné à rendre plus efficace le codage lexical, le seul facteur qui entre en compte est le rendement, qui se calcule en nombre d'ajouts de lemme (avec la classe ou le paradigme de flexion) divisé par le temps de correction.

S'il est possible de mesurer le temps passé sur les ajouts, il est difficile de comparer les scores avec d'autres logiciels. Pour qu'ils soient comparables, il faudrait que le cadre d'expérimentation soit le même. Pour une même langue, le lexique en entrée doit être le même, et le codage doit être fait par plusieurs personnes, par exemple quatre, afin de lisser les différences de connaissances linguistiques. Les personnes participant doivent être natives, avoir une expérience en codage lexical et bien connaître l'outil au moment de

l'expérimentation. Le temps de codage doit être le même, par exemple deux heures. Ensuite, pour éviter de devoir valider les ajouts par plusieurs personnes, seuls les ajouts communs entre trois des quatre personnes sont comptés.

Comme nous n'avons pas les moyens de faire ce genre d'expérience sur l'allemand, nous avons procédé à une évaluation qualitative. Nous avons examiné plus en détail les hypothèses construites avec le seuil cumulatif des terminaisons à 500 et le seuil d'occurrences à 10. Un échantillon avec des exemples de bonnes hypothèses pour chaque classe se trouve en Annexe G, page 263.

La précision est le nombre de tokens proposés qui peuvent améliorer l'analyse divisé par le nombre de tokens proposés. Les tokens qui peuvent améliorer l'analyse ne sont pas simplement des nouvelles entrées dans le lexique, on y compte également les tokens qui indiquent qu'une règle d'analyse est mal codée (des inconnus apparaissent qui ne devraient pas l'être) ou qu'une autre règle d'analyse s'impose (règle de préfixation comme pour les préfixes *un*, *nicht*).

Les résultats sont très bons pour le groupe non capitalisé, où la précision des entrées à considérer est globalement très haute. Nous l'estimons entre 70 % et 85 % pour l'ensemble 3 à 7. Nous l'estimons à une quinzaine de points de moins pour les groupes 8 à 11, mais cette baisse ne gène pas l'efficacité du processus d'enrichissement. Le groupe 11 est un peu particulier, car il est bruité par des racines à une et deux lettres. Il suffit de les filtrer pour obtenir une précision similaire. Le groupe 1 est trop bruité pour être utile, notamment par des mots anglais (*deadline*, *wildlife*, *leadership*, etc.), et latins ou d'origine latine. Ces derniers proviennent de la langue spécialisée médicale, botanique, religieuse, etc. (*labelliformis*, *propediem*, *straminella*, etc.). Nous avons aussi remarqué des noms propres commençant par une minuscule (*iPhone*, *al-Wahid*, *deVille*, etc.). Les tokens allemands sont rares, importants, mais trop longs à trouver dans la masse de données de ce groupe.

La précision est assez basse pour le groupe capitalisé : nous l'estimons à 10 % dans les ensembles 2 et 3. Nous rappelons que les noms ainsi que les nominalisations d'adjectifs et de verbes sont capitalisés en allemand, sans quoi nous aurions pu ignorer les tokens capitalisés.

Ces derniers maux sont globalement les mêmes pour tous les groupes capitalisés. Les résultats sont donc très mauvais, avec une précision extrêmement basse. Les tokens intéressants sont totalement noyés dans la masse des noms propres, quels que soient les paramètres retenus. Les tokens capitalisés les plus faciles à repérer sont les adjectifs capitalisés. Ils sont dérivés d'un nom propre comme les adjectifs de pays ou de nom de personne : *Bayessche*, adjectif de *Bayes*.

9.2.6 Bilan et perspectives

Nous avons expérimenté une méthode originale qui se différencie des autres travaux en ce qu'elle n'essaie pas de reconstituer des paradigmes de flexion. Les résultats sont très satisfaisants pour les verbes et les adjectifs, mais moins pour les noms, qui se trouvent confondus dans une grande masse de noms propres à cause de la capitalisation des noms communs en allemand. La méthode devrait également être expérimentée sur les autres langues, afin de tester sa validité sur son utilité pour la gestion de langues moins flexionnelles. L'outil est utile en l'état, mais peut être rendu plus ergonomique.

Les pistes d'amélioration que nous allons explorer sont les suivantes :

- Lors de la construction de l'hypothèse, définir un nombre minimum de lettres pour la racine, par exemple 3 lettres dont une voyelle comme dans [Nakov et al., 2003].
- Expérimenter l'introduction d'un nombre minimum de lettres pour les terminaisons. Cela les rendra moins linguistiquement motivées, car les terminaisons des noms allemands comptent souvent une seule lettre, mais il est possible que cela augmente la pertinence.
- Faire varier les deux seuils et analyser leurs effets.
- Inverser la logique de présentation : rassembler les mots qui se fléchissent de la même façon car c'est le format le plus pratique et efficace pour la vérification par le linguiste.
- Faire le lien avec les paradigmes de flexion du fléchisseur. Compte tenu du précédent point, des groupes de mots pourront être fléchis en même temps.
- Construire un arbre à partir de la liste de terminaisons résultante et recalculer le nombre d'occurrences des terminaisons en soustrayant le nombre d'occurrences des nœuds pères dont la terminaison est une sous-chaîne. Cela pourrait réduire les ambiguïtés des terminaisons les plus courtes.
- Faire des expériences en séparant les catégories grammaticales, surtout en allemand où la capitalisation des noms communs y incite.
- Evaluer quelle est la précision de proposer comme lemme le mot-forme recensé le plus court dans le groupe des combinaisons <racine ; terminaison>. La proposition d'un lemme accélérera le travail en diminuant les manipulations.
- Changer de corpus. Un corpus moins encyclopédique contiendrait sans doute moins de noms propres.
- Exploiter DBpedia[119], une base de données structurée en triplets RDF, pour filtrer les noms propres. La base contient actuellement les noms de personnes, d'entreprises, géographiques ainsi que les noms de films et d'albums musicaux. Ce filtrage augmentera la précision à la fois pour les tokens non capitalisés (mots anglais), et capitalisés (noms propres).

Pour que les différentes expériences soient comparables, le jeu de données d'origine doit être le même. Or, il est probable de tomber par hasard sur des erreurs au cours des travaux, et il est évidemment difficile de les ignorer ou de simplement les mettre de coté pour plus tard étant donnés l'exploitation commerciale de ces données d'une part, et de possibles meilleurs résultats de l'autre.

9.3 Les erreurs types de l'architecture en extension

Nos travaux sur la construction de la liste des terminaisons avec leurs descriptions nous ont conduit à dresser la typologie d'erreurs suivante. Nous avons d'abord collecté les erreurs, et les avons classées ensuite en quatre niveaux. Tous les exemples sont en allemand, mais la typologie est valable pour toutes les langues. Pour plus de clarté, le lemme (entre parenthèses) n'est pas abrégé.

[119] Voir : http://wiki.dbpedia.org/

Le **niveau 0** est le niveau de la syntaxe du fichier. Comme nous manipulons des fichiers texte, la syntaxe est strictement définie pour délimiter les champs d'informations.

Le **1er niveau** est le niveau des informations individuelles. Le nombre d'entrées du paradigme est correct et l'erreur porte sur la description (1), le lemme (2) ou le mot-forme (3) d'une des entrées. Si elle porte sur la description, elle peut se manifester par le manque d'une ou plusieurs étiquettes (1a), une erreur sur une étiquette (1b) ou par des étiquettes de trop (1c).

1a %Warmwasserheizungen :(Warmwasserheizung)NOM.N.M.P
= absence de la description de composition de mot composé dans toutes les descriptions du paradigme

1a %Serbin :(Serbin)NOM.F.S.HABITANT.HAB_PAYS
%Serbinnen :(Serbin)NOM.F.P.HABITANT.HAB_PAYS
= absence de l'étiquette du cas dans toutes les descriptions du paradigme

1a %Tücher :(Tuch)NOM.A.X
%Tücher :(Tuch)NOM.G.X.P
%Tücher :(Tuch)NOM.N.X.P
%Tüchern :(Tuch)NOM.D.X
= étiquette du pluriel manquant sur deux descriptions du paradigme

1b %Muttis :(Mutti)NOM.A.F.**S**
%Muttis :(Mutti)NOM.D.F.**S**
%Muttis :(Mutti)NOM.G.F.**S**
%Muttis :(Mutti)NOM.N.F.**S**
= le paradigme ne contient que des S pour singulier, or la moitié des mots-formes sont au pluriel

1b %Kasperl :(Kasperl)NOM.A.**M**.P
%Kasperl :(Kasperl)NOM.A.**M**.S
%Kasperl :(Kasperl)NOM.A.**X**.P
%Kasperl :(Kasperl)NOM.A.**X**.S
= le paradigme avec X pour neutre ne devrait pas exister

1b %Nominierung :(Nominierung)NOM.**F**.S.N
%Nominierung :(Nominierung)NOM.**M**.S.A
%Nominierung:(Nominierung)NOM.**M**.S.D
%Nominierungen :(Nominierung)NOM.**F**.P.N
%Nominierungen ::(Nominierung)NOM.**M**.P.A
%Nominierungen ::(Nominierung)NOM.**M**.P.D
= confusion des genres (M et F) au sein du même paradigme

1b %Biodiesel :(Biodiesel)NOM.N.F.S
= le paradigme complet est codé féminin (F) alors que c'est du masculin

1c %Heftung :(Heftung)NOM.A.F.S
%Heftung :(Heftung)NOM.D.F.S
%Heftung :(Heftung)NOM.G.F.S**.MC.NN**
%Heftung :(Heftung)NOM.N.F.S
= étiquettes plausibles en trop sur l'une des entrées

1c %Geranium :(Geranium)NOM.**NPR**.N.X.S
= tout le paradigme contient l'étiquette NPR alors qu'elle est réservé aux noms propres

1c %abflacht :(abflachen)VER.IM.VOUS.VPV.**BVPV**
= la combinaison des étiquettes VPV et BVPV est impossible : si le mot-forme est soit un verbe à particule verbale (VPV), soit une *base de verbe à particule verbale* (BVPV) ; dans le premier cas le mot-forme contient la particule, dans la seconde elle ne le contient pas.

2 %schnä**e**ppre :(-5ppern)VER.SUP.JE
= la forme correcte est *schnäppre*

2 %schnall :(schnallen)VER.PI.TU
= la forme correcte est *schnallst*

2 %Humoristn :(Humorist)NOM.D.M.P
= la forme correcte est Humoristen ; application d'une mauvaise classe de flexion à l'ajout des entrées (fort au lieu de faible).

2 %Mutter**s** :(Mutter)NOM.G.F.S
= un mot-forme au génitif ne peut se terminer par *s*

3 %Klimatisation :(Klimatisatio)NOM.G.F.S
= lemme erroné

3 %Photos :(Foto)NOM.A.M.P
= codage abusif de variation en faisant la liaison par le lemme et non dans le lexique des variantes

3 %Kassia :(Kassi**a**)NOM.D.F.S
%Kassia :(Kassie)NOM.A.F.S
%Kassia :(Kassie)NOM.G.F.S
%Kassia :(Kassie)NOM.N.F.S
= le mauvais encodage des variantes Kassia – Kassie directement dans le lexique fait qu'aucun des paradigmes n'est complet

Le **2e niveau** se situe au niveau des entrées du paradigme. Trois types d'erreurs se présentent : le paradigme est incomplet car il manque une ou plusieurs ambiguïtés sur un mot-forme (1), le paradigme contient des entrées en trop, soit parce qu'il existe des ambiguïtés en trop sur un mot-forme, qui peuvent être incorrectes (2a) ou identiques à des entrées correctes (2b), soit parce que le mot-forme n'existe pas (3).

1 %Studienräte :(Studienrat)NOM.A.M.S
%Studienräte :(Studienrat)NOM.G.M.S
%Studienräten :(Studienrat)NOM.D.M.P
= il manque le nominatif au pluriel

1 %Bolschewikin :(Bolschewikin)NOM.N.F.S
%Bolschewikinnen :(Bolschewikin)NOM.N.F.P
= seul le nominatif (N) est codé, il manque les autres cas : adjectif, génitif et datif

2a %durchblies :(durchblasen)VER.II.IL
%durchblies :(durchblasen)VER.II.IL.VPV
= l'entrée sans l'étiquette VPV est en trop

2b %Argentiniers :(Argentinier)NOM.**G.M.S.**HABITANT.HAB_PAYS
%Argentiniers :(Argentinier)NOM.**M.S.G.**HABITANT.HAB_PAYS
= les étiquettes sont dans un ordre différent

3 %Perestroika :(Perestroika)NOM.A.F.S

%Perestroika :(Perestroika)NOM.D.F.S
%Perestroika :(Perestroika)NOM.G.F.S
%Perestroika :(Perestroika)NOM.N.F.S
%Perestroika**s** :(Perestroika)NOM.G.F.S
= le dernier mot-forme n'existe pas ; le paradigme est complet si on le supprime, car il existe un mot-forme avec la description dans le même paradigme

Le **3e niveau** se situe au niveau du paradigme. Trois situations se présentent : tout un paradigme est codé en trop, avec des informations identiques (1) ou incorrectes (2), ou bien le paradigme est manquant (pas d'exemple).

1 %Katarakt :(Katarakt)NOM.A.F.S
%Katarakt :(Katarakt)NOM.A.**M**.S
%Katarakt :(Katarakt)NOM.D.F.S
%Katarakt :(Katarakt)NOM.D.**M**.S
%Katarakt :(Katarakt)NOM.G.F.S
%Katarakt :(Katarakt)NOM.N.F.S
%Katarakt :(Katarakt)NOM.N.**M**.S
%Katarakte :(Katarakt)NOM.A.F.P
%Katarakte :(Katarakt)NOM.A.**M**.P
%Katarakte :(Katarakt)NOM.G.F.P
%Katarakte :(Katarakt)NOM.G.**M**.P
%Katarakte :(Katarakt)NOM.N.F.P
%Katarakte :(Katarakt)NOM.N.**M**.P
%Katarakten :(Katarakt)NOM.D.F.P
%Katarakten :(Katarakt)NOM.D.**M**.P
%Kataraktes :(Katarakt)NOM.G.**M**.S
%Katarakts :(Katarakt)NOM.G.**M**.S
= les deux paradigmes, masculin (M) et féminin (F) sont complets, mais seul le féminin est correct

2 %Borauit :(Borauit)NOM.A.M.S
%Borauit :(Borauit)NOM.D.M.S
%Borauit :(Borauit)NOM.N.M.S
%Borauite :(Borauit)NOM.A.M.P
%Borauite :(Borauit)NOM.G.M.P
%Borauite :(Borauit)NOM.N.M.P
%Borauiten :(Borauit)NOM.D.M.P
%Borauites :(Borauit)NOM.G.M.S
%Borauits :(Borauit)NOM.G.M.S
= mot inexistant mais parfaitement fléchi

Certaines erreurs peuvent être analysées sur plusieurs niveaux. L'exemple suivant contient simultanément une erreur sur le mot-forme (niv. 1), et sur le lemme (niv. 1). Son paradigme est incomplet car seul le féminin singulier est codé (niv. 2) et le mot-forme et le lemme sont inexistants (niv. 3).

1 %Putzmachererin :(Putzmachererin)NOM.A.F.S
= mot inexistant (confusion avec Putzmacherin), paradigme incomplet

L'exemple suivant combine les deux premiers niveaux, en ce qu'il ajoute un mot-forme inexistant au paradigme bien formé de *Sporthalle* avec une description erronée. Le paradigme contient une entrée de trop (niv.2), et la description contient l'étiquette NPR de trop (niv.1)

2 %Sporthalles :(Sporthalle)NOM.NPR.G.X.S.MC.NN
= mot-forme inexistant, description avec une étiquette en trop

Ces erreurs sont disparates et rarement systématiques. Même quand elles sont systématiques, le nombre d'entrées concernées est limité, comme dans le cas de *Humoristen* où seulement 43 entrées étaient à corriger. Certaines erreurs n'ont pas d'influence du tout sur le système, comme des entrées doubles avec des étiquettes qui ne sont pas dans le même ordre (comme dans l'exemple de *Argentinier*). Ces'informations sont rationalisées pendant la mise sous forme binaire.

Si le nombre d'erreurs reste très limité par rapport au nombre d'informations que contient le lexique, cette typologie d'erreurs est surtout accablante pour la gestion de lexique sous format texte. La plupart des erreurs sont directement liées au formatage en fichiers texte des ressources et la liberté d'édition que cela procure. Il est possible que ces erreurs aient déjà été présentes dans les ressources d'origine à partir desquelles la première acquisition s'est faite, mais toutes peuvent être la conséquence de modifications lexicales erronées.

L'utilisation de l'outil qui donne un accès unifié à toutes les ressources lexicales décrit en 7.2 renforcera la cohérence du lexique, mais la meilleure façon d'éviter ces problèmes est de gérer le lexique sous forme de couples <lemmes-classes de flexion>. Dans ce cas, les erreurs sont toujours systématiques, ce qui aide à les repérer.

9.4 Conclusion

Nous avons encadré le développement des ressources pour une petite dizaine de langues. Les caractéristiques d'une d'entre elles nous a obligé à changer de modèle : alors que toutes les procédures sont prévues pour un encodage en extension, nous avons ajouté des étapes dans la gestion du finnois pour le gérer en intension et de générer le lexique extensionnel ensuite.

Comme nous n'avons pas à disposition les classes de flexion pour les autres langues, nous avons développé un outil pour l'acquisition de lexique qui n'en a pas besoin, ce qui le différencie des approches de l'état de l'art. Les résultats sont honorables, mais peuvent sans aucun doute être améliorés si nous avions à disposition les classes de flexion pour les entrées lexicales. Nous avons constaté en faisant la typologie des erreurs, qu'un bon nombre d'erreurs peuvent être évités avec un codage en intension. La réduction du nombre d'informations à gérer simplifie également la mise à jour lexicale. Nous gagnerons donc sur plusieurs plans avec une conversion en intension des lexiques extensionnels.

Néanmoins, le paradigme de flexion n'est pas une notion linguistique standardisée : selon l'application et la vision linguistique appliquée, il peut y avoir plusieurs découpages en paradigmes pour une même langue, ce qui réduit l'échangeabilité des données.

CONCLUSION ET PERSPECTIVES

Comme pour tout logiciel, le dessein d'un système de traitement automatique des langues évolue avec les objectifs qu'on lui fixe. Il se complexifie souvent lorsque les tâches évoluent ou de nouveaux traitements sont ajoutés. Ses besoins en informations linguistiques augmentent et de nouvelles ressources sont créées. La bonne pratique de séparer traitements et ressources facilite l'intégration de nouvelles langues : le logiciel est rendu générique et les ressources linguistiques contiennent alors les connaissances spécifiques à chaque langue.

L'ajout de traitements et de nouvelles langues crée le problème de la gestion cohérente de toutes les informations que contiennent les ressources linguistiques. Cette thèse a été initiée pour explorer comment faire face aux problèmes de gestion lorsque le système exploite une grande masse d'informations linguistiques qui se trouvent dans des sources hétérogènes.

La question de la gestion des ressources est seulement émergeante aujourd'hui. Le monde industriel semble précurseur à cause des contraintes plus fortes de continuité dans le développement d'une même application et de garantie des performances. Maintenant que les plateformes d'analyse linguistique académiques arrivent à maturité et se stabilisent, nous sommes d'avis que la problématique se posera de façon plus aiguë dans la communauté scientifique.

L'ajout d'un traitement se fait souvent en construisant sur l'existant. Lorsqu'on ajoute des ressources linguistiques, il est primordial de veiller à la cohérence des informations qu'on fournit au système, pourqu'elles ne soient pas contradictoires d'un traitement à l'autre. Les informations peuvent être redondantes à l'intérieur d'une ressource, mais également entre plusieurs ressources, ce qui est une importante source d'erreurs et un casse-tête pour la maintenance. Le formatage même des ressources peut également être à l'origine d'un grand nombre d'erreurs. Le texte formaté notamment a l'avantage d'être très flexible et manipulable, mais présente le grand inconvénient d'être très permissif.

Comme il est difficile, voire parfois impossible, de modifier le système en amont, nous avons construit un environnement de gestion autour de ressources linguistiques existantes qui organise et facilite le travail quotidien de gestion. Si certains outils ont été simples à mettre en place, d'autres ont nécessité une compréhension exhaustive des traitements linguistiques, des informations exploitées et des ressources contenant ces informations.

Pour obtenir cette vue globale, nous avons proposé une méthode pour analyser le fonctionnement d'un système qui intègre des traitements linguistiques. Elle commence par recenser et formaliser les connaissances nécessaires pour chaque traitement linguistique fait par le système. Ensuite, le même recensement des informations que contiennent les ressources linguistiques permet de formaliser le lien entre ressource et traitement. La compréhension du système est complète quand on sait quelles informations sont exploitées par quel traitement et à quel moment.

Cette compréhension profonde du système est indispensable pour concevoir les outils qui composent l'environnement de gestion. Un premier groupe d'outils s'adresse au responsable de la gestion : ils permettent la planification des travaux à court et à long terme. Ils assurent un suivi de l'évolution des lexiques, une comparaison entre différentes versions, un traçage

des modifications sur la durée, un état des lieux et l'estimation de la complexité des grammaires.

Un second groupe d'outils se destine au linguiste pour le codage des ressources et les tests et validations des traitements. Ces outils optimisent le temps de travail linguistique qui est souvent laborieux à cause de la grande masse de données à gérer. Le plus important parmi ces outils est un prototype qui est en cours d'industrialisation : il donne un accès unique à toutes les ressources lexicales d'une langue et donne la possibilité de vérifier si les éléments lexicaux utilisés dans les grammaires existent bien dans les lexiques. Les informations lexicales qui sont physiquement dispersées sur plusieurs sources sont recoupées pour éviter toute redondance. Ainsi on garde la flexibilité des fichiers texte qui permet des manipulations à grande comme à petite échelle, demande un faible coût en terme d'outils d'édition et est archivable à long terme sans poser de problème technologique. Cette interface d'édition restreint la permissivité, car plusieurs aspects sont contrôlés lors de l'importation ou de l'enregistrement.

La masse de données aurait pu être minimale, si l'architecture lexicale utilisait des paradigmes de flexion pour coder le lexique. Nous avons appliqué cette architecture pour le finnois, dont le lexique extensionnel compte quelques millions d'entrées à cause de la nature agglutinative de la langue. Le choix du codage en extension comme architecture de référence est lié à la façon dont les autres lexiques ont été constitués : elle privilégie l'intégration de lexiques existants. La conversion en lexique intensionnel est à prévoir, car le codage en intension simplifie le codage lexical et réduit les coûts de gestion à long terme.

Les paradigmes de flexion sont actuellement un élément manquant pour développer des outils de mise à jour lexicale efficaces. Même si nous avons obtenu de bons résultats avec l'outil d'acquisition que nous avons développé et expérimenté sur l'allemand, cet outil demande un paramétrage peu ergonomique. L'utilisation des paradigmes réduit considérablement la masse de données lexicales, ainsi que le nombre d'erreurs. Pour ne pas toucher à l'architecture existante, nous l'instaurerons comme une étape préalable au processus actuel. L'architecture lexicale que nous avons mise en place pour le finnois deviendrait progressivement l'architecture dominante.

En plus de l'évolution vers des lexiques intensionnels, nous voyons comme perspectives l'amélioration des outils mis en place. L'outil donnant un accès unifié à tous les lexiques est en cours d'industrialisation, son utilité étant prouvée. Le recodage garantit la maintenance et l'optimisation du logiciel. L'outil d'enrichissement lexical manque pour l'instant d'ergonomie, mais nous avons démontré qu'il est possible de construire un tel outil sans avoir à disposition les classes de flexion.

Il est important d'explorer d'autres stratégies d'acquisition. Comme celle que nous appliquons repose majoritairement sur des ressources existantes, nous n'avons pas de modèle pour les langues *peu dotées*. L'inégalité au niveau des ressources linguistiques entre les langues reflète les inégalités économiques actuelles. Plus que de sauver des langues, il s'agit de réduire la fracture numérique qui existe indéniablement. Nos expériences avec l'outil d'acquisition sont aussi un premier pas dans ce sens, et nous espérons pouvoir mener des expériences sur d'autres langues dans un avenir proche.

En ce qui concerne la prédictibilité de l'adaptation des ressources à un certain type de données entrantes, tout reste à faire. Si nous savons construire des modèles différents, nous ne disposons aujourd'hui d'aucune mesure qui nous indique si notre modèle est suffisant pour les types de données rencontrés. Avec la multiplication des canaux d'information, la publication de fichiers audio et vidéo s'est généralisée et nous assistons à une multimédiatisation de notre environnement. Cela nous amène à explorer de nouveaux modèles pour le traitement du son et

de l'image. S'il existe des solutions pour la transcription automatique et pour la génération de métadonnées, les méthodes d'analyse textuelle ne sont pas habituées à ce que les informations en entrée présentent un certain pourcentage d'erreurs dû à la transcription automatique. Les paradigmes connus sont plutôt issus des textes littéraires ou journalistiques. Le traitement des appels téléphoniques et des conversations ne sont que quelques exemples de types de documents qui nous posent des défis, notamment celui entre la généricité et la spécialisation des ressources (voir Annexe J). En effet, l'utilisation des mêmes ressources pour traiter des types de texte très différents mène souvent à des résultats décevants. L'adaptation des ressources linguistiques aux nouveaux types de textes et l'efficacité des méthodes existantes constituent déjà nos axes de recherche principaux.

BIBLIOGRAPHIE

[Abbès et Boualem, 2008] R. Abbès et M. Boualem. Dissymétrie entre l'indexation et la recherche d'information en langue arabe. In *Actes de TALN 2008*, Avignon, 2008.

[ACE, 2008] ACE. ACE: Automatic Content Extraction, 2008. http://www.nist.gov/speech/tests/ace/, consulté en ligne le 8/08/08.

[Adolphs, 2008] Peter Adolphs. Acquiring a poor man's inflectional lexicon for german. In *Proceedings of LREC'08*, Marrakech, Morocco, 2008.

[Agirre et al., 2010] Eneko Agirre, Giorgio Maria Di Nunzio, Thomas Mandl, et Arantxa Otegi. Clef 2009 ad hoc track overview: Robust-wsd task. *Lecture Notes in Computer Science*, 6241/2010: 36–49, 2010.

[Amsler, 1980] R. A. Amsler. *The Structure of the Merriam-Webster Pocket Dictionary*. PhD thesis, University of Texas, 1980.

[Anis, 2002] Jacques Anis. Communication électronique scripturale et formes langagières. In *Actes de RHRT 4 : Réseaux Humains / Réseaux Technolotiques*, Poitiers, 2002.

[ANS, 2005] ANS. E-ANS, Algemene Nederlandse Spraakkunst, versie 1.2. En ligne, 2005. http://www.let.ru.nl/ans/e-ans/, consulté en ligne le 11/11/08.

[Antoni-Lay et al., 1994] M.-H. Antoni-Lay, G. Francopoulo, et L. Zaysser. A generic model for reusable lexicons: The genelex project. *Literary and Linguistic Computing*, 9(1): 47–54, 1994.

[Artemenko et al., 2006] O. Artemenko, T. Mandl, M. Shramko, et C. Womser-Hacker. Evaluation of a language identification system for mono- and multi-lingual text documents. In ACM (ed.), *Proceedings of the 2006 ACM Symposium on Applied Computing SAC '06 (Dijon)*, pages 859–860., New York, 2006. doi: 10.1145/1141277.1141473. URL du poster associé : http://eprints.rclis.org/archive/00007081/ consulté le 5/09/08.

[Atkins, 1991] B. T. S. Atkins. Building a lexicon: The contribution of lexicography. *International Journal of Lexicography*, 4(3): 167–204, 1991.

[Baayen et al., 1995] R. H. Baayen, R. Piepenbrock, et L. Gulikers. The CELEX Lexical Database (CD-ROM). Technical report, Linguistic Data Consortium (University of Pennsylvania), Phildadelphia, PA, 1995.

[Banko et Brill, 2001] M. Banko et E. Brill. Mitigating the paucity-of-data problem: Exploring the effect of training corpus size on classifier performance for natural processing. In *Proceedings of HLT 2001*, 2001.

[Banks et Sundaram, 1990] J. S. Banks et R. K. Sundaram. Repeated games, finite automata and complexity. *Games and Economic Behaviour*, 2: 97–117, 1990.

[Baroni et al., 2004] M. Baroni, S. Bernardini, F. Comastri, L. Piccioni, A. Volpi, G. Aston, et M. Mazzoleni. Introducing the la repubblica corpus: A large, annotated, TEI(XML)-compliant corpus of newspaper italian. In *Proceedings of LREC 2004*, pages 1771–1774, Lisbon, 2004. ELDA.

[Biber, 2004] Douglas Biber. Conversation text types: A multi-dimensional analysis. In Gérald Purnelle, Cédrick Fairon, et Anne Dister (ed.), *Le poids des mots: Actes de JADT 2004*, pages 15–34, Louvain, 2004. Presses universitaires de Louvain.

[Bonhomme et Lopez, 2000] Patrick Bonhomme et Patrice Lopez. TagML : XML encoding of resources for lexicalized tree adjoining grammars. In *Proceedings of LREC 2000*, Athens, 2000.

[Booch et al., 1998] Grady Booch, James Rumbaugh, et Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison Wesley, Reading, Massachusetts, September 1998.

[Booij, 1994] Geert E. Booij. *Yearbook of Morphology 1993*, chapter Against split morphology, pages 27–50. Springer-Verlag New York, LLC, 1994.

[Bozzi et al., 2009] Laurent Bozzi, Philippe Suignard, et Claire Waast-Richard. Segmentation et classification non supervisée de conversations téléphoniques automatiquement retranscrites. In *Actes de TALN 2009*, 2009.

[Brill, 1992] Eric Brill. A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied Natural Language Processing*, Trento, Italy, 1992. doi: 10.3115/974499.974526.

[Brill, 1995] Eric Brill. Transfomation-based error-driven learning and natural language processing : A case study in part of speech tagging. *Computational Linguistics*, 21(4): 543–565, 1995.

[Broeder et al., 2001] Daan Broeder, Freddy Offenga, Don Willems, et Peter Wittenburg. The IMDI metadata set, its tools and accessible linguistic databases. In S. Bird, P. Buneman, et M. Liberman (ed.), *Proceedings of the IRCS Workshop on Linguistic Databases*, pages 48–55, Philadelphia, 2001.

[Broeder et al., 2004] Daan Broeder, Thierry Declerck, Laurent Romary, Eric De La Clergerie, Sven Uneson, Markus Strömqvist, et Peter Wittenburg. A large metadata domain for language resources. In *Proceedings of the 4th International Conference on Language Resources and Evaluation - LREC'04*, pages 369–372, 2004.

[Burnard, 2000] Lou Burnard. Where did we go wrong? a retrospective look at the british national corpus. In B. Kettemann et G. Marko (ed.), *Language and Computers, Teaching and Learning by doing Corpus Analysis. Proceedings of the Fourth International Conference on Teaching and Language Corpora*, pages 51–70, Rodopi, 2000.

[Burnard, 2001] S. Bauman S. DeRose Burnard, L. (ed.). *Guidelines for Text Encoding and Interchange*. Text Encoding Initiative, Chicago and Oxford, 2001.

[Buscaldi et Rosso, 2008] Davide Buscaldi et Paolo Rosso. QA with a disambiguated document collection. In *Working Notes for the CLEF 2008 Workshop, 17-19 September*, Aarhus, Denmark, 2008.

[Cahill, 2001] Lynne Cahill. Semi-automatic construction of multilingual lexicons. *Machine Translation Review*, 12: 67–74, December 2001.

[Cahill et Gazdar, 1999] Lynne Cahill et Gerald Gazdar. The PolyLex architecture: multilingual lexicons for related languages. *Traitement automatique des langues*, 40:2: 5–23, 1999.

[Cailliau, 2006] Frederik Cailliau. Un modèle pour unifier la gestion de ressources linguistiques en contexte multilingue. In *Actes de TALN 2006*, Leuven, 2006.

[Cailliau et Giraudel, 2008] Frederik Cailliau et Aude Giraudel. Enhanced search and navigation on conversational speech. In *Proceedings of SSCS 2008*, Singapore, 2008. SIGIR 2008.

[Cailliau et Loupy, 2007] Frederik Cailliau et Claude de Loupy. Aides à la navigation dans un corpus de transcriptions d'oral. In *Actes de TALN 2007*, Toulouse, 2007.

[Cailliau et al., 2008] Frederik Cailliau, Aude Giraudel, et Céline Poudat. Fusion de ressources hétérogènes pour la recherche d'information. In *Actes de Coria 2008*, pages 433–440, Trégastel, 2008.

[Calvet, 1987] Louis-Jean Calvet. *Politique linguistique et impérialisme; l'Institut Linguistique d'Eté*, chapter Chapitre 14 : La guerre des langues et les politiques linguistiques, pages 205–217. Payot, 1987. Version anglaise :

http://people.bu.edu/manfred/CalvetCh14anglaisSIL.pdf, consulté en ligne le 8/08/08.

[Carlos et al., 2009] Cohan Sujay Carlos, Monojit Choudhury, et Sandipan Dandapat. Large-coverage root lexicon extraction for Hindi. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 121–129, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[Cavnar et Trenkle, 1994] William B. Cavnar et John M. Trenkle. N-gram-based text categorization. In *Proceedings of the Third Symposium on Document Analysis and Info*, pages 161–175, Las Vegas, 1994. NV: UNLV Publications/Reprographics.

[Chalvin et Mangeot, 2006] Antoine Chalvin et Mathieu Mangeot. Méthodes et outils du GDEF. In *Proceedings of EURALEX 2006*, Turin, 2006.

[Church, 2004] Ken Church. Speech and language processing: Can we use the past to predict the future? In P. Sojka, I. Kopeček, et K. Pala (ed.), *Proceedings of Text, Speech and Dialogue,TSD 2004*. Springer, 2004.

[Cieri et Liberman, 2008] Christopher Cieri et Mark Liberman. 15 years of language resource creation and sharing: A progress report on LDC activities. In *Proceedings of LREC 2008*, Marrakech, 2008.

[Clément et al., 2004] Lionel Clément, Benoît Sagot, et Bernard Lang. Morphology based automatic acquisition of large-coverage lexica. In *Proceedings of LREC'04*, Lisbonne, 2004.

[Courtois, 1990] Blandine Courtois. Dictionnaire par classes flexionnelles delas-cv, delas-cn, delas-ca (version v06-2). Technical report, Laboratoire D'Automatique Documentaire Et Linguistique, Univ. de Paris 7, 1990.

[Courtois, 1995] Blandine Courtois. Buts et méthodes de l'élaboration des dictionnaires électroniques du ladl. In *Cahier du CIEL, 1994-1995*, pages 87–107, Université Paris VII, 1995.

[Crestan et Loupy, 2004] Eric Crestan et Claude de Loupy. Browsing help for a faster retrieval. In *Proceedings of COLING 2004*, pages 576–582, Genève, Suisse, 2004.

[Creutz et al., 2007] Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, et Andreas Stolcke. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Trans. Speech Lang. Process.*, 5 (1): 1–29, 2007. ISSN 1550-4875. doi: http://doi.acm.org/10.1145/1322391.1322394.

[Cunningham et al., 2002] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, et Valentin Tablan. GATE: an architecture for development of robust hlt applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[Cunningham et al., 2009] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Marin Dimitrov, Mike Dowman, Niraj Aswani, Ian Roberts, Yaoyong Li, Adam Funk, Genevieve Gorrell, Johann Petrak, Horacio Saggion, Danica Damljanovic, et Angus Roberts. *Developing Language Processing Components with GATE Version 5 (a User Guide). For GATE version 5.1-snapshot (development builds) (built October 26, 2009)*. The University of Sheffield, Sheffield, 2009. Consulté en ligne 9/09/09.

[Daelemans et al., 1992] Walter Daelemans, Koenraad De Smedt, et Gerald Gazdar. Inheritance in natural language processing. *Comput. Linguist.*, 18 (2): 205–218, 1992. ISSN 0891-2017.

[Dal et al., 2004] Georgette Dal, Nabil Hathout, et Fiammetta Namer. Morphologie constructionnelle et traitement automatique des langues: le projet MORTAL. *Lexique*, 16 (Corbin, Pierre): 199– 229, 2004. Villeneuve d'Ascq: Presses Universitaires du Septentrion.

[Dutoit et al., 2003] Dominique Dutoit, Pierre Nugues, et Patrick de Torcy. The integral dictionary: A lexical network based on componential semantics. *Proceedings of the 2003 International Conference on Computational Science and its Applications, ICCSA 2003, Lecture Notes in Computer Science*, 2667: 368–377, May 2003. doi: 10.1007/3-540-44839-X.

[EAGLES, 1993] EAGLES. EAGLES Guidelines, 1993.

[Emerson, 2000] Thomas Emerson. Segmenting chinese in unicode. In *Proceedings of the 16th International Unicode Conference*, Amsterdam, 2000.

[Erjavec et al., 2000] Tomaz Erjavec, Roger Evans, Nancy Ide, et Adam Kilgarriff. The concede model for lexical databases. In *Proceedings of the Second Conference on Language Resources and Evaluation*, pages 355–362, 2000.

[Evans, 1998] A.S. Evans. Reasoning with UML class diagrams. In *Workshop on Industrial Strength Formal Methods, WIFT'98*, Florida, 1998. IEEE Press.

[Evans et Gazdar, 1989] Roger Evans et Gerald Gazdar. Inference in DATR. In *Proceedings of the fourth conference on European chapter of the Association for Computational Linguistics*, pages 66–71, Morristown, NJ, USA, 1989. Association for Computational Linguistics. doi: 10.3115/976815.976824.

[Evans et Gazdar, 1996] Roger Evans et Gerald Gazdar. DATR: A language for lexical knowledge representation. *Computational Linguistics*, 22.2: 167–216, 1996.

[Evens et Smith, 1979] Martha Evens et Raoul Smith. A lexicon for a computer question-answering system. *American Journal of Computational Linguistics*, 4: Microfiche 83: 1–96, 1979.

[Fairon et al., 2006] Cédrick Fairon, Jean Klein, et Sébastien Paumier. *Le langage SMS. Etude d'un corpus informatisé à partir de l'enquête 'Faites don de vos SMS à la science'*, volume 3.1 of *Cahiers du Cental*. Presses universitaires de Louvain, Louvain-la-Neuve, 2006.

[Fayet-Scribe, 1997] Sylvie Fayet-Scribe. Chronologie des supports, des dispositifs spatiaux, des outils de repérage de l'information. *Solaris*, 4: En ligne, 1997. En ligne. Consulté le 8/08/08.

[Fellbaum, 1998] Christiane Fellbaum (ed.). *WordNet. An Electronic Lexical Database*. The MIT Press, 1998.

[Ferdeghini et Niggi, 2001] Marina Ferdeghini et Paola Niggi. *Le Robert & Nathan. Grammaire de l'italien*. Nathan/VUEF, Paris, 2001. ISBN 2 09 181108-4.

[Ferrucci et Lally, 2004] David Ferrucci et Adam Lally. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10 (3-4): 327–348, 2004. doi: 10.1017/S1351324904003523.

[Ferrucci et al., 2006] David Ferrucci, Adam Lally, Daniel Gruhl, Edward Epstein, Marshall Schor, J. William Murdock, Andy Frenkiel, Eric W. Brown, Thomas Hampp, Yurdaer Doganata, Christopher Welty, Lisa Amini, Galina Kofman, Lev Kozakov, et Yosi Mass. Towards an interoperability standard for text and multi-modal analytics. IBM Research Report RC24122 (W0611-188), IBM Research Division, 2006.

[Fersøe, 2004] Hanne Fersøe. Validation manual for lexica, v2.0. report submitted to ELRA under the validation unit contract elra/0209/val-1. Technical report, ELRA VCOM, January 2004.

[Fersøe et Olsen, 2005] Hanne Fersøe et Sussi Olsen. Methodology for a quick quality check for WLR-lexica v2.0. Technical report, ELRA VCOM, October 2005.

[Flickinger et al., 1985] Daniel Flickinger, Carl Pollard, et Thomas Wasow. Structure-sharing in lexical representation. In *Proceedings of the 23rd annual meeting on Association for Computational Linguistics*, pages 262–267, Morristown, NJ, USA, 1985. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/981210.981242.

[Florian et al., 2003] Radu Florian, Abe Ittycheriah, Hongyan Jing, et Tong Zhang. Named entity recognition through classifier combination. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 168–171, Morristown, NJ, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119176.1119201.

[Fox et Fox, 2002] Brian Fox et Christopher. J. Fox. Efficient stemmer generation. *Information Processing Management*, 38 (4): 547–558, July 2002. doi: 10.1016/S0306-4573(01)00047-4.

[Fradet et al., 1999] Pascal Fradet, Daniel Le Métayer, et Michaël Périn. Consistency checking for multiple view software architectures. *SIGSOFT Softw. Eng. Notes*, 24 (6): 410–428, 1999. ISSN 0163-5948. doi: 10.1145/318774.319258.

[France et al., 1998] Robert B. France, Andy S. Evans, Kevin C. Lano, et Bernhard Rumpe. The UML as a formal modeling notation. *Computer Standards & Interfacing*, 19 (7): 325–334, 1998.

[Francopoulo, 2004] Gil Francopoulo. Proposition de norme des lexiques pour le traitement automatique du langage. Technical Report 1.10, Projet Normalangue-RNIL, 13 mai 2004 2004.

[Francopoulo et al., 1992] Gil Francopoulo, Didier Massot, et Geneviève Morize. *Guide d'utilisation des fonctions du poste de travail lexicographique AlethGD*. GSI-ERLI, 1.2 edition, juin 1992. http://web.archive.org/web/20051031200003/http://www.uhb.fr/langues/balneo/html/inv9451.htm.

[Gagnon, 1998] Damien Gagnon. La maturité syntaxique. *Correspondance*, 3 (4): en ligne : http://www.ccdmd.qc.ca/correspo/Corr3–4/Maturite.html, 1998.

[Gallaher, 2007] Carolyn Gallaher. The role of protestant missionaires in Mexico's indigenous awakening. *Bulletin of Latin American Research*, 26 (1): 88–111, 2007.

[Genelex, 1993] Consortium Genelex. Rapport sur la couche syntaxique. Technical Report 4.0, Projet EUREKA Genelex, 1993.

[Genelex, 1994] Consortium Genelex. Rapport sur la couche morphologique. Technical Report 3.3, Projet EUREKA Genelex, November 1994.

[Genelex, 1995] Consortium Genelex. Rapport sur la couche sémantique. Technical Report 2.1, Projet EUREKA Genelex, 1995.

[Gijsel et al., 2005] Sofie Van Gijsel, Dirk Speelman, et Dirk Geeraerts. A variationist, corpus linguistic analysis of lexical richness. In *Proceedings of Corpus Linguistics 2005*, Birmingham, UK, 2005.

[Goldsmith, 2001] John Goldsmith. Unsupervised learning of the morphology of a natural language. *Comput. Linguist.*, 27 (2): 153–198, 2001. ISSN 0891-2017. doi: http://dx.doi.org/10.1162/089120101750300490.

[Gonnet et Tompa, 1987] Gaston H. Gonnet et Frank W. Tompa. Mind your grammar: a new approach to modelling text. In *Proceedings of Very Large Data Bases*, pages 339–346, 1987.

[Grefenstette et Tapanainen, 1994] Gregory Grefenstette et Pasi Tapanainen. What is a word, what is a sentence? Problems of tokenization. In *Proceedings of the 3rd International Conference on Computational Lexicography and Text Research. COMPLEX '94*, pages 79–87, Budapest, 1994.

[Gross, 1988] Gaston Gross. Degré de figement des noms composés. *Langages*, 90: 57–70, 1988.

[Gross, 1968] Maurice. Gross. *Grammaire transformationnelle du français. Vol. 1, Syntaxe du verbe*, volume 1. Larousse, Paris, 1968.

[Gross, 1989] Maurice Gross. The use of finite automata in the lexical representation of natural language. *Lecture Notes in Computer Science*, 377: 34–50, 1989.

[Guerra, 2006] Guillaume Guerra. Studying how to structure linguistic data. Rapport de stage., Sinequa - Ensimag Grenoble, 2006.

[Guimier de Neef et al., 2007] Emilie Guimier de Neef, Arnaud Debeurme, et Jungyeul Park. TILT correcteur de SMS : évaluation et bilan quantitatif. In *Actes de TALN*, page 123–132, Toulouse, 2007.

[Guyot et al., 2008] Jacques Guyot, Gilles Falquet, Saïd Radhouani, et Karim Benzineb. UNIGE experiments on Robust Word Sense Disambiguation. In *Working Notes for the CLEF 2008 Workshop, 17-19 September*, Aarhus, Denmark., 2008.

[Habert, 2005] Benoît Habert. *Instruments et ressources électroniques pour le français*. L'essentiel français. Ophrys, Gap/Paris, 2005.

[Hajič, 2000] Jan Hajič. Morphological tagging: data vs. dictionaries. In *Proceedings of the First Conference on North American Chapter of the Association for Computational Linguistics*, volume 4 of *ACM International Conference Proceeding Series*, pages 94–101, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

[Harman, 1988] Donna Harman. Towards interactive query expansion. In *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 321–331, New York, NY, USA, 1988. ACM. ISBN 2-7061-0309-4. doi: 10.1145/62437.62469.

[Harman, 1991] Donna Harman. How effective is suffixing? *Journal of the American Society for Information Science*, 42: 7–15, 1991.

[Harman, 2000] Donna Harman. What we have learned, and not learned, from TREC. In *BCS IRSG '2000 Proceedings*, pages 2–20, 2000.

[Harris, 1964] Zellig Harris. Transformations in linguistic structure. In *Proceedings of the American Philosophical Society*, volume 108, pages 418–122, 1964.

[Heinecke et al., 2008] Johannes Heinecke, Grégory Smits, Guimier De Neef Emilie Chardenon, Christine and, Estelle Maillebuau, et Malek Boualem. TiLT : plate-forme pour le traitement automatique des langues naturelles. *TAL*, 49-2: 17–41, 2008.

[Humphreys, 1996] Lee Humphreys. Use of linguistic resources like translation memories in machine translation systems. In *Proceedings of the EAMT Workshop, TKE'96*, pages 29 – 30, Vienna, Austria, 1996.

[Ide et Sperberg-McQueen, 1995] Nancy Ide et C.M. Sperberg-McQueen. *The Text Encoding Initiative: Background and Context*, chapter The Text Encoding Initiative: Its History, Goals, and Future Development, pages 5–15. Kluwer Academic Publishers, Dordrecht, 1995.

[Ide et Véronis, 1993] Nancy Ide et Jean Véronis. Extracting knowledge bases from machine-readable dictionaries: Have we wasted our time? In *Knowledge Bases & Knowledge Structures 93*, Tokyo, 1993.

[Ide et Véronis, 1994] Nancy Ide et Jean Véronis. MULTEXT: Multilingual text tools and corpora. In COLING'94 (ed.), *Proceedings of the 15th Conference on Computational Linguistics*, volume 1, pages 588–592, Morristown, NJ, 1994. COLING'94 (Kyoto, Japan). doi: 10.3115/991886.991990.

[Ide et Véronis, 1995a] Nancy Ide et Jean Véronis. Knowledge extraction from machine-readable dictionaries: An evaluation. In P. Steffens (ed.), *Machine Translation and the Lexicon. Proceedings of the third international EAMT Workshop (Heidelberg 1993).*, volume 898 of *Lecture Notes In Computer Science*, pages 19–34, London, 1995. Springer-Verlag.

[Ide et Véronis, 1995b] Nancy Ide et Jean Véronis. Encoding dictionaries. *Computers and the Humanities*, 29: 167–180, 1995.

[Ide et Véronis, 1998] Nancy Ide et Jean Véronis. Word Sense Disambiguation: The state of the art. *Computational Linguistics*, 24:1: 1–40., 1998.

[Ide et al., 1992] Nancy Ide, Jean Véronis, Susan Warwick-Armstrong, et Nicoletta Calzolari. Principles for encoding machine readable dictionaries. In H. Tommola, K. Varantola, T. Salmi-Tolonen, et Y. Schopp (ed.), *EURALEX'92 Proceedings*, volume 2 of *Studia Translatologica*, pages 239–246, Tampere, Finland, 1992.

[Ide et al., 2000] Nancy Ide, Adam Kilgarriff, et Laurent Romary. A formal model of dictionary structure and content. In *Proceedings of EURALEX 2000*, page 113–126, Stuttgart, 2000.

[IMDI, 2001a] IMDI. Metadata elements for session descriptions. Technical report, MPI Nijmegen, 2001. Version 2.5 (June 2001).

[IMDI, 2001b] IMDI. Mapping IMDI session descriptions with OLAC. Technical report, MPI Nijmegen, 2001. Version 1.04 (August 2001).

[IMDI, 2003] IMDI. Metadata elements for session descriptions. Technical report, MPI Nijmegen, 2003. Version 3.0.4 (October 2003).

[ISO-TC37/SC4, 2008] ISO-TC37/SC4. ISO/TC 37/SC 4 N453 (N330 Rev.16) Language resource management — Lexical markup framework (LMF) Gestion de ressources linguistiques — Cadre de balisage lexical (LMF). ISO FDIS 24613:2008, March 2008.

[Jelinek, 2004] Frederick Jelinek. Some of my best friends are linguists. In *Proceedings of LREC 2004*, 2004.

[Joffe et de Schryver, 2004] David Joffe et Gilles-Maurice de Schryver. TshwaneLex – a state-of-the-art dictionary compilation program. In *Proceedings of the Eleventh EURALEX International Congress, EURALEX 2004*, page 99–104, 2004.

[Järvinen et al., 2004] Timo Järvinen, Laari Mikko, Timo Lahtinen, Paajanen. Sirkku, Paljakka Pirkko, Soininen Mirkka, et Tapanainen Pasi. Robust language analysis components for practical applications. In *Proceedings of the COLING Workshop Robust and Adaptive Information Processing for Mobile Speech Interfaces*, Genève, Suisse, 2004.

[Kalai et Stanford, 1988] E. Kalai et W. Stanford. Finite rationality and interpersonal complexity in repeated games. *Econometrica*, 56: 397–410, 1988.

[Kaplan, 1982] Joan Kaplan, Robert et Bresnan. *The Mental Representation of Grammatical Relations*, chapter Lexical-functional grammar: a formal system for grammatical representation, page 173–281. MIT Press, Cambridge, Massachusetts, 1982.

[Karlsson et al., 1994] Fred Karlsson, Atro Voutilainen, Juha Heikkilä, et Arto Anttila. *Constraint Grammar: A Language-Independent Formalism for Parsing Unrestricted Text*. Mouton de Gruyter, 1994.

[Kayser, 1998] Daniel Kayser. *La représentation des connaissances*. Hermès, Paris, 1998.

[Kilgarriff, 1997] Adam Kilgarriff. I don't believe in word senses. *Computers and the Humanities*, 31: 91–113, 1997.

[Kilgarriff, 1999] Adam Kilgarriff. Generic encoding principles. Technical report, CONCEDE project, December 1999.

[Kilgarriff, 2003] Adam Kilgarriff. No-bureaucracy evaluation. In *Proceedings of the Workshop on Evaluations Initiatives in NLP (EACL)*, Budapest, Hongrie, 2003. Information Technology Research Institute, University of Brighton (ITRI-03-18).

[Kilgarriff, 2007] Adam Kilgarriff. Googleology is bad science. *Computational Linguistics*, 33 (1): 147–151, 2007. ISSN 0891-2017. doi: 10.1162/coli.2007.33.1.147.

[Kilgarriff et al., 2004] Adam Kilgarriff, Pavel Rychlý, Pavel Smrž, et David Tugwell. The sketch engine. In *Proceedings of Euralex*, Lorient, 2004.

[Kobus et al., 2008] Catherine Kobus, François Yvon, et Géraldine Damnati. Normalizing SMS: are two metaphors better than one ? In *Proceedings of COLING 2008*, pages 441–448, Manchester, 2008.

[Koskenniemi, 1983] Kimmo Koskenniemi. *Two-level morphology: A general computational model for word-form recognition and production*. PhD thesis, University of Helsinki, Department of General Linguistics, 1983.

[Langer, 2002] Stefan Langer. Grenzen der Sprachenidentifizierung. In *Tagungsband KONVENS 2002*, pages 99–106, Saarbrücken, 2002.

[Lee et Sabourian, 2007] Jihong Lee et Hamid Sabourian. Coase theorem, complexity and transaction costs. *Journal of Economic Theory*, 135: 214–235, 2007.

[Leech et al., 1994] Geoffrey Leech, Roger Garside, et Michael Bryant. *Corpus-based research into Language*, chapter The large-scale grammatical tagging of text: experience with the British National Corpus, pages 47–63. Rodopi, Amsterdam, 1994.

[Lenci et al., 2000] Alessandro Lenci, Nuria Bel, Federica Busa, Nicoletta Calzolari, Elisabetta Gola, Monica Monachini, Antoine Ogonowski, Ivonne Peters, Wim Peters, Nilda Ruimy, Marta Villegas, et Antonio Zampolli. SIMPLE: A general framework for the development of multilingual lexicons. *International Journal of Lexicography*, 13(4): 249–263, 2000. doi: 10.1093/ijl/13.4.249.

[LIM, 1994] LIM. GSI-ERLI. Language Industry Monitor, Sept-Oct 1994.

[LIPN-Paris13, 2007] LIPN-Paris13. Typologie textuelle : Etat de l'art et applications (lot 2.1). Technical report, Projet Textcoop, 2007.

[Llamas Pombo, 2001] Elena Llamas Pombo. La construction visuelle de la parole dans le livre médiéval. *Diogène*, 4-196: 40–52, 2001. ISSN 0419-1633.

[Loupy, 2000] Claude de Loupy. *Évaluation de l'apport de connaissances linguistiques en désambiguïsation sémantique et recherche documentaire*. Thèse de doctorat en informatique, Université d'Avignon et des Pays de Vaucluse, 2000.

[Loupy et Crestan, 2004] Claude de Loupy et Eric Crestan. *Systèmes de recherche d'information*, chapter Traitement automatique des langues et systèmes de recherche d'information. Éditions Hermès, Paris, 2004.

[Loupy et Gonçalves, 2008] Claude de Loupy et Sandra Gonçalves. Aide à la construction de lexiques morphosyntaxiques. In *Proceedings of EURALEX 2008*, Barcelone, Espagne, 2008.

[Loupy et al., 2009] Claude de Loupy, Michaël Bagur, et Helena Blancafort. Association automatique de lemmes et de paradigmes de flexion à un mot inconnu. In *Actes de TALN 2009*, 2009.

[Lovins, 1968] Julie Beth Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11: 22–31, 1968.

[Mandl et al., 2006] Thomas Mandl, Margaryta Shramko, Olga Tartakovski, et Christa Womser-Hacker. Language identification in multi-lingual web-documents. In *Proceedings of NLDB*, pages 153–163, 2006. doi: 10.1007/11765448_14.

[Manfredi, 2007] Victor Manfredi. Bailey-bridge to oil doom: Kay williamson reveals s.i.l.'s official role in the maladministration of the post-biafran niger delta. http://people.bu.edu/manfredi/WilliamsonSIL.pdf, 2007.

[Mangeot et al., 2003] Mathieu Mangeot, Gilles Sérasset, et Mathieu Lafourcade. Construction collaborative de données lexicales multilingues, le projet Papillon. *Traitement Automatique des Langues*, 44:2: 151–176, 2003.

[Mariani, 1995] Joseph Mariani. Mlap SPEECHDAT project. deliverable 3.2.2. relations of a european center for spoken language resources (ECSLR) with on-going projects. Technical report, LIMSI-CNRS, 1995.

[Martínez-Santiago et al., 2008] Fernando Martínez-Santiago, José M. Perea-Ortega, et Miguel Angel García-Cumbreras. SINAI at Robust WSD Task @ CLEF 2008: When WSD is a good idea for information retrieval tasks? In *Working Notes for the CLEF 2008 Workshop, 17-19 September*, Aarhus, Denmark, 2008.

[Maynard et al., 2001] Diana Maynard, Valentin Tablan, Cristian Ursu, Hamish Cunningham, et Yorick Wilks. Named entity recognition from diverse text types. In *Recent Advances in Natural Language Processing 2001 Conference*, Tzigov Chark, Bulgaria, 2001.

[Mel'čuk, 1973] Igor A. Mel'čuk. *Trends in Soviet Theoretical Linguistics*, chapter Towards a Linguistic 'Meaning <=> Text' Model, pages 33–57. D. Reidel, Dordrecht - Holland, 1973.

[Mel'čuk et al., 1995] Igor Aleksandrovič Mel'čuk, André Clas, et Alain Polguère. *Introduction à la lexicologie explicative et combinatoire*. Duculot, Louvain-la-Neuve, 1995.

[Mohri, 1997] Mehryar Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23 (2): 269–311, Jun. 1997. ISSN 0891-2017.

[Monachini et al., 2003] Monica Monachini, Francesca Bertagna, Nicoletta Calzolari, Nancy Underwood, et Costanza Navarretta. Towards a standard for the creation of lexica. Technical report, ELRA-VCOM, Paris, 2003.

[Moreau, 2006] Fabienne Moreau. *Revisiter le couplage traitement automatique des langues et recherche d'information*. PhD thesis, Université de Rennes, 2006.

[Mota et Santos, 2008] Cristina Mota et Diana Santos. Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O segundo HAREM. In *Actas do Encontro do Segundo HAREM (Aveiro)*. Linguateca, 2008.

[Moulinier et al., 2001] Isabelle Moulinier, J. Andrew McCulloh, et Elizabeth Lund. West Group at CLEF 2000: Non-english monolingual retrieval. In C. Peters (ed.), *Revised Papers From the Workshop of Cross-Language Evaluation Forum on Cross-Language information Retrieval and Evaluation*, volume 2069 of *Lecture Notes in Computer Science*, pages 253–260, London, 2001. Springer-Verlag.

[MUC7, 1997] MUC7 (ed.). *Proceedings of the Message Understanding Conference 7, MUC-7*, 1997.

[Nakov et al., 2003] Preslav Nakov, Yury Bonev, Galia Angelova, Evelyn Gius, et Walter von Hahn. Guessing morphological classes of unknown german nouns. In *Proceedings of RANLP'03*, pages 319–326, Borovetz, Bulgaria, 2003.

[Navarro et al., 2008] Sergio Navarro, Fernando Llopis, et Rafael Muñoz. IRn in the CLEF Robust WSD Task 2008. In *Working Notes for the CLEF 2008 Workshop, 17-19 September, Aarhus, Denmark.*, 2008.

[Neuhaus, 1986] H. Joachim Neuhaus. Lexical database design: The shakespeare dictionary model. In *Proceedings of the 11th International Conference on Computational Linguistics*, pages . 441–444., Bonn, 1986. IPK.

[Nevis, 2000] Joel Ashmore Nevis. *Morphology. An International handbook on inflection and word-formation*, volume 1, chapter 41. Clitics, pages 388–404. Walter de Gruyter, Berlin - New York, 2000.

[New et al., 2004] Boris New, Christophe Pallier, Marc Brysbaert, et Ludovic Ferrand. Lexique 2 : A new french lexical database. *Behavior Research Methods, Instruments, & Computers*, 36 (3): 516–524, 2004.

[New et al., 2007] Boris New, Marc Brysbaert, Jean Véronis, et Christophe Pallier. The use of film subtitles to estimate word frequencies. *Applied Psycholinguistics*, 28: 661–677, 2007.

[Nguyen et Boitet, 2007] Hong-Thai Nguyen et Christian Boitet. Vers un méta-EDL complet, puis un EDL universel pour la TAO. In *Actes de TALN 2007*, Toulouse, 2007.

[Oliver et Tadic, 2004] Antoni Oliver et Marko Tadic. Enlarging the Croatian morphological lexicon by automatic lexical acquisition from raw corpora. In *Proceedings of LREC'04*, 2004.

[Otegi et al., 2008] Arantxa Otegi, Eneko Agirre, et German Rigau. IXA at CLEF 2008 Robust-WSD Task: using Word Sense Disambiguation for (Cross Lingual) Information Retrieval. In *Working Notes for the CLEF 2008 Workshop, 17-19 September*, Aarhus, Denmark, 2008.

[Paprotté et Schumacher, 1993] Wolf Paprotté et Frank Schumacher. MULTILEX – final report WP 9: MLEXd. report MWP 8 – MS. Technical report, MULTILEX, Münster, 1993.

[Paroubek et Rajman, 2000] Patrick Paroubek et Martin Rajman. *Ingénierie des Connaissances*, chapter Etiquetage morpho-syntaxique. Hermès Science Publications, Paris, 2000.

[Paulin, 1999] René Paulin. Education aux médias : France Inter, 1999. http://www.cndp.fr/archivage/valid/669/669-451-472.pdf, consulté en ligne le 8/08/2008.

[Paumier, 2002] Sébastien Paumier. *Manuel d'utilisation d'Unitex*. Université de Marne-la-Vallée, 2002.

[Pédauque, 2006] Roger T. Pédauque. *Le document à la lumière du numérique*. C&F Editions, 2006. ISBN 2-915825-04-1.

[Petasis et al., 2002] George Petasis, Vangelis Karkaletsis, George Paliouras, Ion Androutsopoulos, et Costas D. Spyropoulos. Ellogon: A new text engineering platform. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pages 72–78, Las Palmas, Spain, 2002.

[Piskorski et al., 2007] Jakub Piskorski, Marcin Sydow, et Anna Kupsc. Lemmatization of polish person names. In *BSNLP Workshop at ACL*, Prague, 2007.

[Poibeau et Kosseim, 2001] Thierry Poibeau et Leila Kosseim. Proper name extraction from non-journalistic texts. In Walter Daelemans, Khalil Sima'an, Jorn Veenstra, et Jakub Zavrel (ed.), *Computational Linguistics in the Netherlands. Selected Papers from the Eleventh CLIN Meeting*, pages 144–157, Amsterdam/New York, 2001.

[Polguère, 2003] Alain Polguère. *Lexicologie et sémantique lexicale*. Presses de l'Université de Montréal, Montréal, Qué, 2003.

[Porter, 1980] Martin F. Porter. An algorithm for suffix stripping. *Program*, 14: 130–137, 1980.

[Prager, 1999] John M. Prager. Linguini: language identification for multilingual documents. *Journal of Management Information System*, 16, 3: 71–101, Décembre 1999.

[Prószéky, 1994] Gabor Prószéky. Industrial applications of unification morphology. In *Proceedings of the Fourth Conference on Applied Natural Language Processing (Stuttgart, Germany, October 13 - 15, 1994). Applied Natural Language Conferences.*, pages 213–214, San Francisco, CA, 1994. Morgan Kaufmann Publishers, San Francisco, CA, 213-214. DOI= http://dx.doi.org/10.3115/974358.974415. doi: 10.3115/974358.974415.

[Pustejovsky, 1991] James Pustejovsky. The generative lexicon. *Computational Linguistics*, 17 (4): 409–441, 1991.

[Pustejovsky, 1995] James Pustejovsky. *The Generative Lexicon*. The MIT Press, Cambridge, MA, 1995.

[Pustejovsky, 2001] James Pustejovsky. *The language of word meaning*, chapter Type construction and the logic of concepts, pages 91–123. Cambridge University Press, Cambridge, 2001. Revision of : Pustejovsky, J. (1998), "Specification of a Top Concept Lattice", ms. Brandeis University.

[Richardson et al., 1998] Stephen D. Richardson, William B. Dolan, et Lucy Vanderwende. MindNet: acquiring and structuring semantic information from text. In *Proceedings of the 17th international Conference on Computational Linguistics*, volume 2, pages 1098–1102, Morristown, NY, 1998. doi: 10.3115/980432.980749.

[Riegel et al., 1999] Martin Riegel, Jean-Christophe Pelat, et René Rioul. *Grammaire méthodique du français*. Presses Universitaires de France, 1999.

[Rijsbergen, 2006] Keith van Rijsbergen. Introduction to some models in information retrieval. Présentation à EARIA 2006, Grenoble, 2006.

[Roche et Shabes, 1997] Emmanuel Roche et Yves Shabes (ed.). *Finite-State Language Processing*. MIT Press, Cambridge, MA, USA, 1997. ISBN 0262181827.

[Rolling, 1993] Loll Rolling. E.C. language projects. In *Proceedings of MT Summit IV, International Cooperation for Global Communication*, pages 97–116, Kobe, Japan, 1993.

[Romary et al., 2004] Laurent Romary, Susanne Salmon-Alt, et Gil Francopoulo. Standards going concrete : from LMF to Morphalou. In *Proceedings of Workshop on Electronic Dictionaries (Coling 2004)*, Geneva, Switzerland, 2004.

[Sagot, 2005] Benoît Sagot. Automatic acquisition of a slovak lexicon from a raw corpus. In *Proceedings of TSD'05*, Lecture Notes in Artificial Intelligence 3658, pages 156–163, Karlovy Vary, République Tchèque, 2005. Springer-Verlag.

[Sagot, 2007] Benoît Sagot. Building a morphosyntactic lexicon and a pre-syntactic processing chain for polish. In *Proceedings of LTC 2007*, Poznań, Pologne, 2007.

[Schuurman et al., 2003] Ineke Schuurman, Machteld Schouppe, Ton van der Wouden, et Heleen Hoekstra. CGN, an annotated corpus of spoken Dutch. In Anne Abeillé, Silvia Hansen-Schirra, et Hans Uszkoreit (ed.), *Proceedings of 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*, pages 101–108, Budapest, 2003.

[Schuurman et al., 2004] Ineke Schuurman, Wim Goedertier, Heleen Hoekstra, Nelleke Oostdijk, Richard Piepenbrock, et Machteld Schouppe. Linguistic annotation of the Spoken Dutch Corpus: If we had to do it all over again. In *Proceedings of the IV International Conference on Language Resources and Evaluation, vol. I, pp. 57-60.*, Lisbonne, 2004.

[Seite et al., 1992] Bernard Seite, Daniel Bachut, D Maret, et Brigitte Roudaud. Presentation of the EUROLANG project. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 4 (Nantes)*, pages 1289–1293, Morristown, NJ, 1992. doi: 10.3115/992424.992485.

[Shriberg, 2005] Elisabeth Shriberg. Spontaneous speech: How people really talk, and why engineers should care. In *Proceedings of Eurospeech*, pages 1781–1784, 2005.

[Silberztein, 1993] Max Silberztein. *Dictionnaires électroniques et analyse automatique de textes: le système INTEX*. Masson, Paris, 1993.

[Silberztein, 2003] Max Silberztein. *NooJ manual*. Université de Franche-Comté, 2003.

[Silberztein, 2005] Max Silberztein. NooJ's dictionaries. In *Proceedings of the 2nd Language and Technology Conference*. Poznan University, 2005.

[Simons et Bird, 2003] Gary F. Simons et Steven Bird. OLAC metadata. Technical report, Open Language Archives Community, 2003.

[SIMPLE Consortium, 2000] SIMPLE Consortium. SIMPLE linguistic specifications. LE-SIMPLE (LE4-8346), Deliverable d2.1. Technical report, ILC and University of Pisa, Pisa, 2000. Lenci, A. and Busa, F. and Ruimy, N. and Gola, E. and Monachini, M. and Calzolari, N. and Zampolli, A. et al.

[Sperberg-McQueen et Burnard, 1994] C. M. Sperberg-McQueen et L. Burnard (ed.). *Guidelines for Text Encoding and Interchange*. Text Encoding Initiative, Chicago and Oxford, 1994.

[Sérasset, 1993] Gilles Sérasset. Recent trends of electronic dictionary research and development in europe. Technical memorandum, EDR, Tokyo, Japan, 1993.

[Sérasset, 1994] Gilles Sérasset. *Sublim : un systeme universel de bases lexicales multilingues et Nadia : sa specialisation aux bases lexicales interlingues par acceptions*. PhD thesis, Université Joseph-Fourier - Grenoble I, Grenoble, 1994.

[Surcin, 2008] Sylvain Surcin. Evaluation de l'intégration des grammaires de text dans un système de Question-Réponses. livrable Textcoop, lot 4.3. Technical report, Sinequa Labs, 2008.

[Swanson, 1988] Don R. Swanson. Historical note: information retrieval and the future of an illusion. *Journal of the American Society for Information Science*, 39 (2): 73 – 145, March 1988. doi: 10.1002/(SICI)1097-4571(198803)39:2<92::AID-ASI4>3.0.CO;2-P.

[Tadic et Fulgosi, 2003] Marko Tadic et Sanja Fulgosi. Building the Croatian morphological lexicon. Ín *MorphSlav '03: Proceedings of the 2003 EACL Workshop on Morphological Processing of Slavic Languages*, pages 41–46, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[Takebayashi, 1993] Yoichi Takebayashi. EDR electronic dictionary. In *Proceedings of MT Summit IV, International Cooperation for Global Communication*, pages 97–116, Kobe, Japan, 1993.

[The Unicode Consortium, 2007] The Unicode Consortium. *The Unicode Standard, Version 5.1.0, defined by: The Unicode Standard, Version 5.0, as amended by Unicode 5.1.0*. Addison-Wesley, Boston, MA, 2007.

[Tjong Kim Sang, 2002] Erik F. Tjong Kim Sang. Introduction to the CoNLL-2002 shared task: language-independent named entity recognition. In *COLING-02: proceedings of the 6th conference on Natural language learning*, pages 1–4, Morristown, NJ, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118853.1118877.

[Tjong Kim Sang et De Meulder, 2003] Erik F. Tjong Kim Sang et Fien De Meulder. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 142–147, Morristown, NJ, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119176.1119195.

[Tompa, 1997] Frank W. Tompa. Views of text. Invited speaker at Digital Media Information Base (DMIB'97), Nara, Japan,, November 1997.

[Tutin et Véronis, 1998] Agnès Tutin et Jean Véronis. Electronic dictionary encoding: Customizing the TEI guidelines. In *Proceedings of Euralex 1998*, pages 4–8, Liège, 1998.

[UHB, 2001] UHB. Regroupement de tous les logiciels : Station genelex. En ligne. Consulté le 8/08/08, 2001. http://www.uhb.fr/langues/balneo/html/inv9451.htm. Accessible via http://web.archive.org/.

[VanRullen et al., 2005] Tristan VanRullen, Philippe Blache, Cristel Portes, Stéphane Rauzy, Jean-François Maeyhieux, Marie-Laure Guénot, Jean-Marie Balfourier, et Emmanuel Bellengier. Une plateforme pour l'acquisition, la maintenance et la validation de ressources lexicales. In *Actes de TALN-05*, Dourdan, 2005.

[Volk et al., 2002] Martin Volk, Bärbel Ripplinger, Spela Vintar, Paul Buitelaar, Diana Raileanu, et Bogdan Sacaleanu. Semantic annotation for concept-based cross-language medical information retrieval. *International Journal of Medical Informatics*, 67 (1/3): 79–112, 2002.

[Vossen, 1998] Piek Vossen (ed.). *EuroWordNet: a multilingual database with lexical semantic networks*. Kluwer Academic Publishers, Norwell, MA, USA, 1998. ISBN 0-7923-5295-5.

[Vossen, 2002] Piek Vossen. EuroWordNet: General document. Technical report, EuroWordNet: Project LE2-4003 & LE4-8328 report, 2002.

[Véronis et Khouri, 1995] Jean Véronis et Liliane Khouri. Etiquetage grammatical multilingue: le projet Multext. *Traitement Automatique des Langues*, 36 (1/2): 233–248, 1995.

[Šnajder et al., 2008] Jan Šnajder, Bojana Dalbelo Bašic, et Marko Tadic. Áutomatic acquisition of inflectional lexica for morphological normalisation. *Information Processing and Management*, 44: 1720–1731, 2008. ISSN 0306-4573. doi: 10.1016/j.ipm.2008.03.006.

[Weiss, 2005] Dawid Weiss. Stempelator: A hybrid stemmer for the Polish language. technical report RA-002/05. Technical report, Institute of Computing Science, Poznań University of Technology, Poland, 2005.

[Widlöcher et Bilhaut, 2005] Antoine Widlöcher et Frédérik Bilhaut. La plate-forme Linguastream : un outil d'exploration linguistique sur corpus. In *Actes de la 12e Conf´erence Traitement Automatique du Langage Naturel (TALN)*, Dourdan, France, 2005.

[Wittenburg, 2001] Peter Wittenburg. Lexical structures. Technical report, MPI Nijmegen, 2001.

[Wittenburg et al., 2002] Peter Wittenburg, Wim Peters, et Sebastian Drude. Analysis of lexical structures from field linguistics and language engineering. In *Proceedings of LREC2002*, Las Palmas, 2002.

[Wittenburg et al., 2004] Peter Wittenburg, Daan Broeder, Richard Piepenbrock, et Kees van de Veer. Databases for linguistic purposes: a case study of being always too early and too late. In *Proceedings of the E-MELD (Electronic Metastructure for Endangered Languages Data) Workshop*, 2004.

[Woordenlijst, 2005] Woordenlijst. Woordenlijst der Nederlandse Taal. En ligne, 2005. http://woordenlijst.org, consulté le 11/11/08.

[Zanchetta et Baroni, 2005] Eros Zanchetta et Marco Baroni. Morph-it! a free corpus-based morphological resource for the italian language. In *Proceedings of Corpus Linguistics*, Birmingham, 2005.

[Zeman, 2008] Daniel Zeman. *Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers*, chapter Unsupervised Acquiring of Morphological Paradigms from Tokenized Text, pages 892–899. Springer-Verlag, Berlin, Heidelberg, 2008. ISBN 978-3-540-85759-4. doi: 10.1007/978-3-540-85760-0_114.

ANNEXES

Annexe A. Liste des projets de recherche auxquels nous avons participé.

Voici une liste des projets de recherche auquel nous avons participé avec un bref descriptif. Parmi les partenaires, le leader est indiqué par un asterisk.

Blogoscopie

Type : ANR

Début : 2007

Durée : 2 ans

Partenaires : LINA* (Nantes-Atlantique), Overblog, Sinequa

Description : Analyse d'image et de tendance sur les blogs. La blogosphère étant devenue une source d'information importante, l'analyse des opinions exprimées intéressent beaucoup d'acteurs différents. Sinequa s'est investi dans la constitution de communautés de blogs ayant les mêmes centres d'intérêt et dans la détection des opinions en développant un lexique-grammaire.

Site web : http://www.blogoscopie.org/

Infom@gic ST2.31, « Callsurf »

Type : ANR

Début : 2006

Durée : 3 ans

Partenaires: EDF R&D, Limsi, Sinequa, Temis, Vecsys*

Description : Fouille de données sur des conversations agent-client dans un centre d'appel. Sinequa a développé un modèle de language pour le langage conversationnel et développé l'IHM multimodale qui s'est greffé sur le moteur de recherche de Sinequa. Ce prototype intègre les apports de chaque partenaire, notamment les transcriptions automatiques et enregistrements, et la segmentation structurelle et thématique des conversations.

Lirics

Type : Projet européen

Début : 2006

Durée : 2 ans

Partenaires : De très nombreux partenaires universitaires.

Description : Lirics a rassemblé au niveau international tous les acteurs, essentiellement académiques, à l'origine de la norme ISO. La liste des 21 industriels du « Industrial Advisory Group », dont faisait partie Sinequa, n'a pas été rendue publique. Le suivi officiel était plutôt réduit (deux réunions), mais une liste de mails permettait un suivi plus régulier de l'avancement des travaux et des discussions en cours.

Site web : http://lirics.loria.fr/

Normalangue-RNIL

Type : ANR (Technolangue)

Début : 2002

Durée : 3 ans

Partenaires : INRIA-Loria*, Afnor et de nombreux partenaires français, universitaires et industriels, dont Sinequa.

Description : « Ressources Normalisées en Ingénierie Linguistique ». En 2002 fut créé le comité TC37/SC4 à l'ISO avec le but de proposer une norme pour le formatage des ressources linguistiques en TAL et des opérations d'annotation afin de promouvoir l'échange de données entre les acteurs du domaine. RNIL a permis de mettre en place le comité français et d'établir une proposition de normalisation française qui sera reprise par la suite au niveau mondial à l'ISO.

Site web : http://www.technolangue.net/article.php3?id_article=82

Oural

Type : ANR

Début : 28/11/2003

Durée : 2 ans

Partenaires : LIA (Avignon), LIP6 (Paris 6), LPE (Paris 5), SILEX (Lille 3), Sinequa*, Valoria (Bretagne-Sud)

Description : OURAL, *OUtil et Ressource pour l'Analyse de la Langue*, avait pour but de mettre à disposition de la communauté des outils et des ressources de base pour le traitement de la langue écrite ou parlé. Les outils ont la particularité de mixer à la fois des évaluations probabilistes et des automates déterministes.

Site web : http://www.technolangue.net/article.php3?id_article=79

Piithie

Type : ANR

Début : 2007

Durée : 2 ans

Partenaires : Advestigo, LIA (Avignon), LINA (Nantes-Atlantique), Sinequa*

Description : Détection de plagiat et suivi d'impact. Amélioration de la recherche de documents similaires selon des degrés variables, avec sélection des candidats sur internet ou en local à partir de l'empreinte lexicale d'un document.

Site web : http://www.piithie.com/

Vodel

Type : ANR

Début : 2005

Durée : 2 ans

Partenaires : Cismef (CHU Rouen), EADS-DS, LASELDI (Franche-Comté), LITIS (INSA-Rouen), Mémodata*

Durée : 2 ans

Description : Vodel, *Valorisation Ontologique des Dictionnaires Electroniques*.

Résultats pour Sinequa : « Fusion » des données Mémodata avec les données sémantiques de Sinequa pour améliorer le filtrage sémantique dans Intuition. Application à la recherche interlingue et à la navigation avancée en intégrant le service Sensagent de Mémodata.

Site web : http://vodel.insa-rouen.fr/ (plus en ligne)

Textcoop

Type : ANR

Début : 2005

Durée : 3 ans

Partenaires : Sinequa*, LIPN (Paris 13), IRIT (Toulouse)

Description : Amélioration de la recherche d'information en utilisant les typologies textuelles et de grammaires de textes Les typologies permettent d'appliquer des grammaires adaptées aux différents types de textes afin de déterminer la structure de ceux-ci et donc de savoir quelle partie contient quel type d'information. Une telle connaissance permet à un moteur de recherche ou un système de question/réponse de déterminer de manière plus fine où se trouve l'information pertinente.

Site web : http://www.textcoop.org/

TSSRC

Type : ANR, projet blanc

Début : 2006

Durée : 3 ans

Partenaires : LDI* (Paris 13), Sinequa

Description : « Typologie sémantique systématique des relations causales ». Description et détection automatique de structures causales dans des corpus journalistiques afin de structurer automatiquement les textes présentés au lecteur.

Annexe B. Découpage en mots par un moteur de recherche français

La copie d'écran ci-dessous vient d'une nouvelle publiée sur le site Clubic[120] le 3 juillet 2008. On voit que la requête de *loutre* donne comme premier résultat le site du gouvernement français qui s'occupe de l'Outre-Mer.

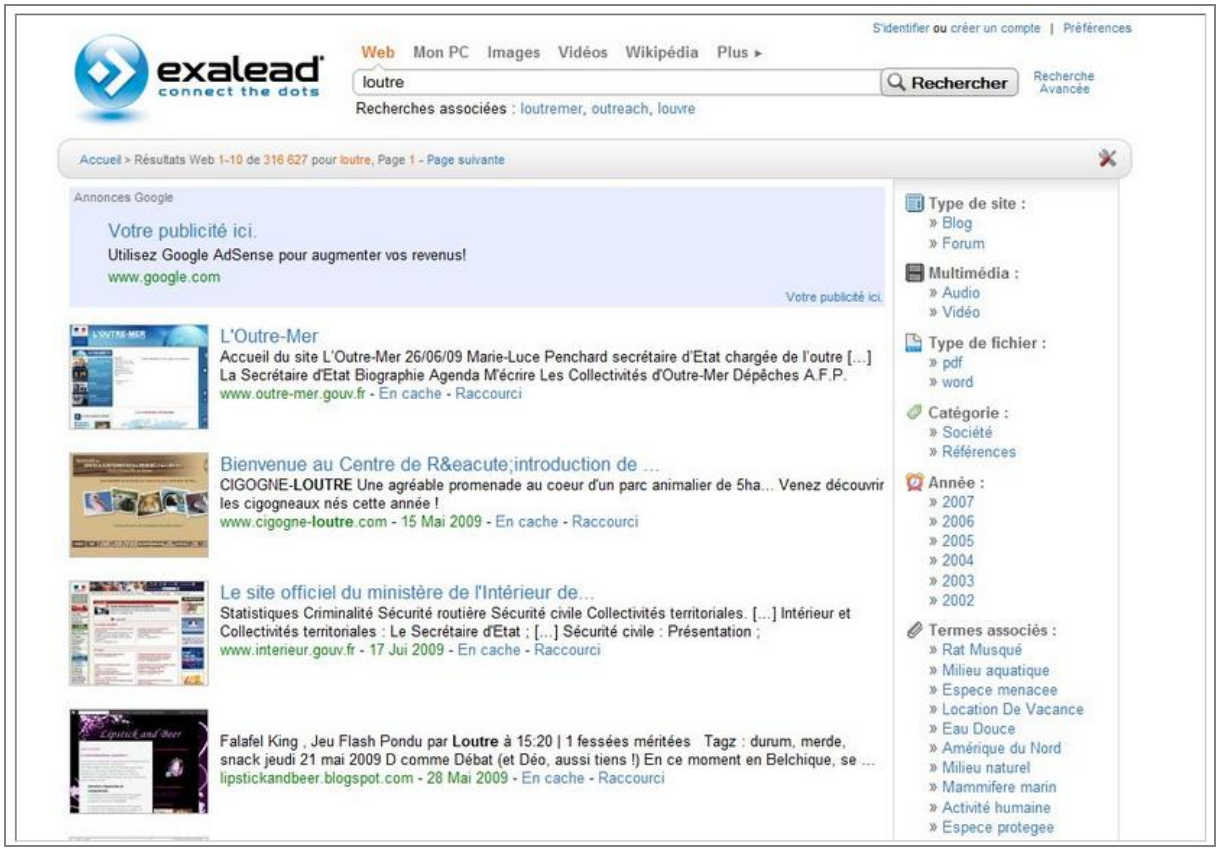


Figure 81 : La recherche de "loutre" sur un moteur de recherche français (3/07/08)

Un mois plus tard, les résultats sur le site en ligne n'étaient plus les mêmes, comme nous pouvons voir sur la copie d'écran ci-dessous (datant du 8/8/09). Néanmoins, comme « Recherches associées », le moteur propose toujours « Loutre Mer ».

[120] http://www.clubic.com/actualite-286148-exalead-prepare-refonte-moteur-recherche.html

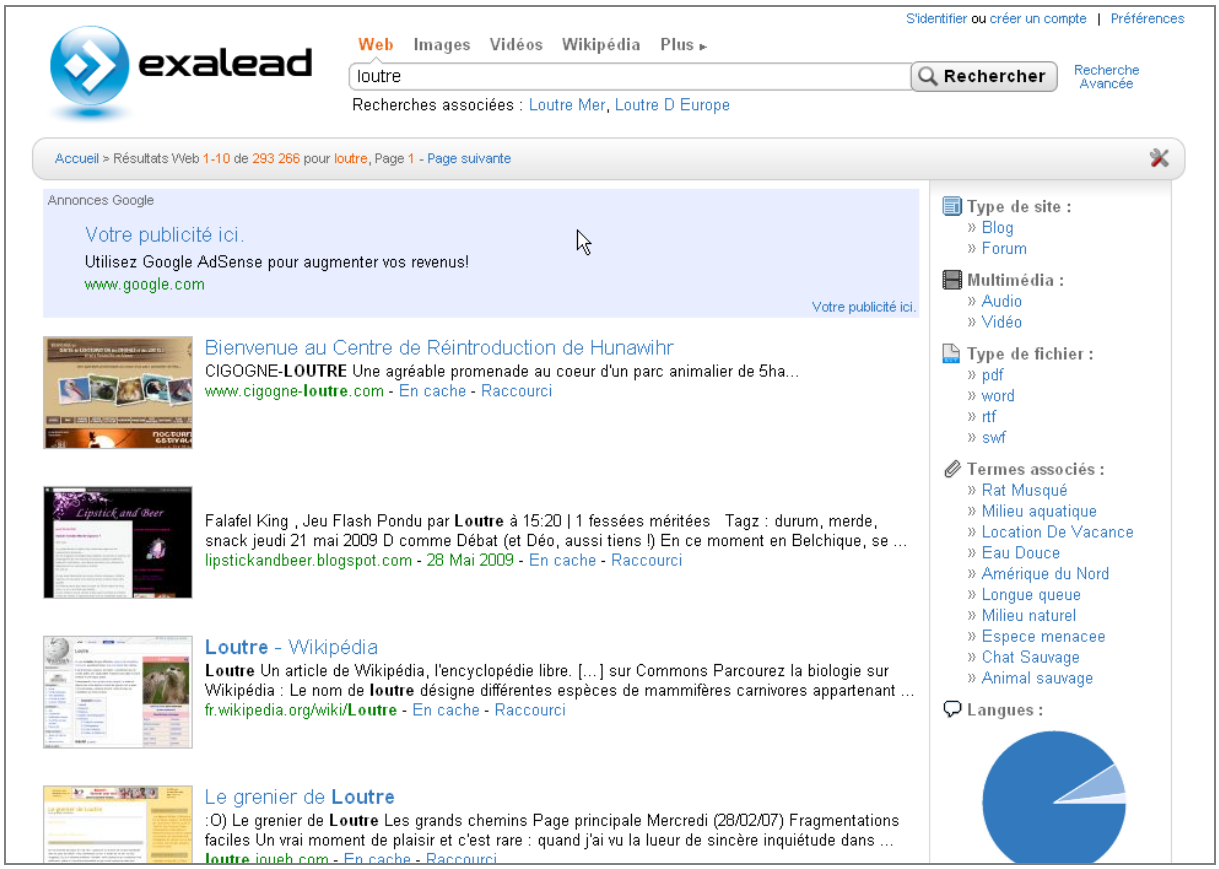


Figure 82 : La recherche de "loutre" sur un moteur de recherche français (8/08/09)

Notons tout de même que vérification faite un an plus tard (4/08/10), le moteur traite l'apostrophe en français de façon correcte.

Annexe C. The *Boring* Couple

Figure 83 : Exemple de la difficulté de la détection d'entités (ITWire du 2/07/08)

L'article ci-dessus est un exemple concret de l'ambiguïté que peuvent porter les noms de personne, ce qui est la difficulté principale rencontrée à la détection automatique.

Annexe D. Schémas relationnels préliminaires

Plusieurs schémas relationnels ont précédé le schéma final qui a été implémenté dans la base de connaissances lexicales (LKB) présenté dans 7.2.3, p. 163. Les trois schémas ci-dessous sont le fruit d'une réflexion sur la conception du modèle relationnel. Le premier schéma est le schéma le plus récent des trois. Il a largement inspiré le schéma final présenté dans la figure 49 (p. 165). Il s'inscrit dans une approche minimaliste, car il représente les informations après sélection des fonctionnalités essentielles pour notre prototype.

Le deuxième schéma (figure 85) est assez complexe et illustre la complexité du schéma si on essaie d'implémenter toutes les fonctionnalités possibles liées à la gestion lexicale. On voit sur le côté droit notamment des caractérisations du mot comme mot simple, abréviation, mot composé, etc. qui n'ont pas été retenues dans l'application. En haut à gauche se trouvent les entités de gestion dont les informations n'existent pas actuellement, comme le nom de l'auteur, le nom du valideur, le type de modification, etc. Nous avons considéré que ces informations ne sont finalement pas essentielles pour notre application, mais peuvent l'être dans d'autres situations. Le dernier schéma (figure 86) est une rationalisation fonctionnelle du deuxième schéma qui laisse de coté les méta-données de gestion qui peuvent être importantes dans certains cadres, mais ne le sont pas dans le nôtre. La transition entre le troisième et le premier schéma est le résultat d'une rationalisation plus technique que fonctionnelle.

Les premières expérimentations avec ces schémas ont été réalisées sous linux avec MySQL, des scripts PERL qui alimentent la base, et phpMyAdmin[121]. Le rendu graphique des schémas a été réalisé avec le logiciel Druid[122].

[121] Voir : http://www.phpmyadmin.net/.
[122] *Druid* est un éditeur graphqiue de base de données : http://druid.sourceforge.net/.

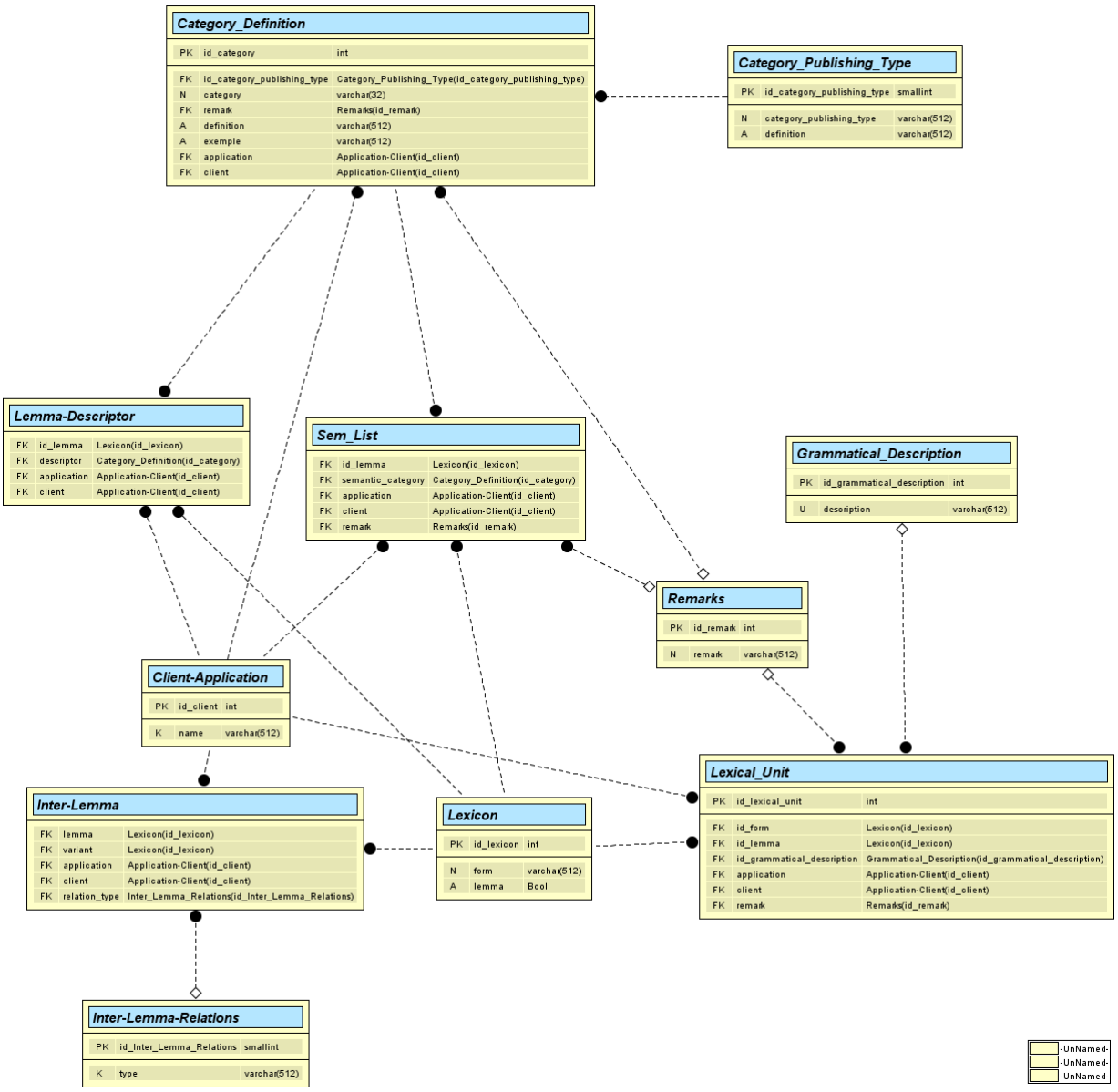


Figure 84 : Schéma relationnel minimaliste 4.0

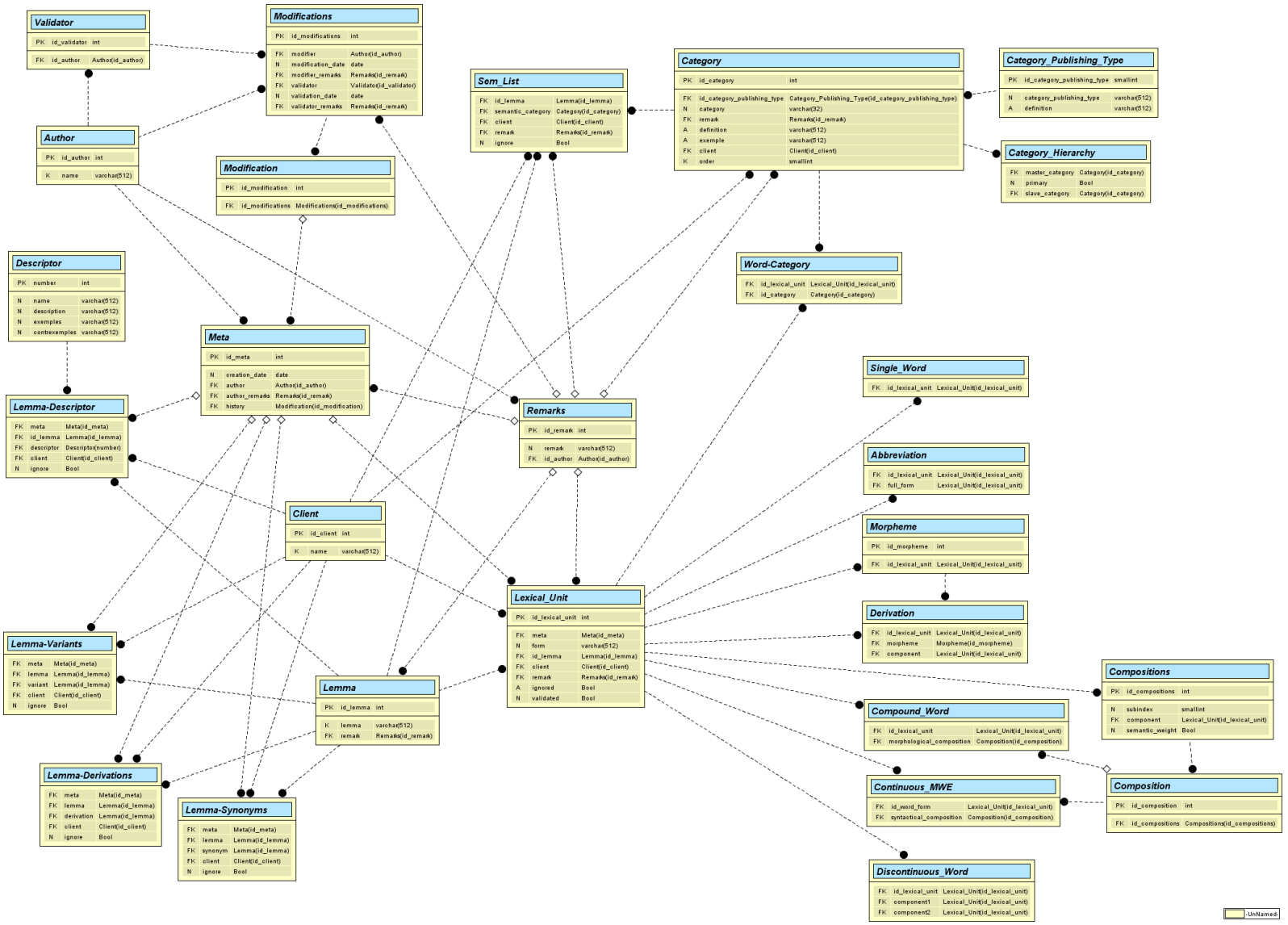


Figure 85 : Schéma relationnel 3.0 « complet »

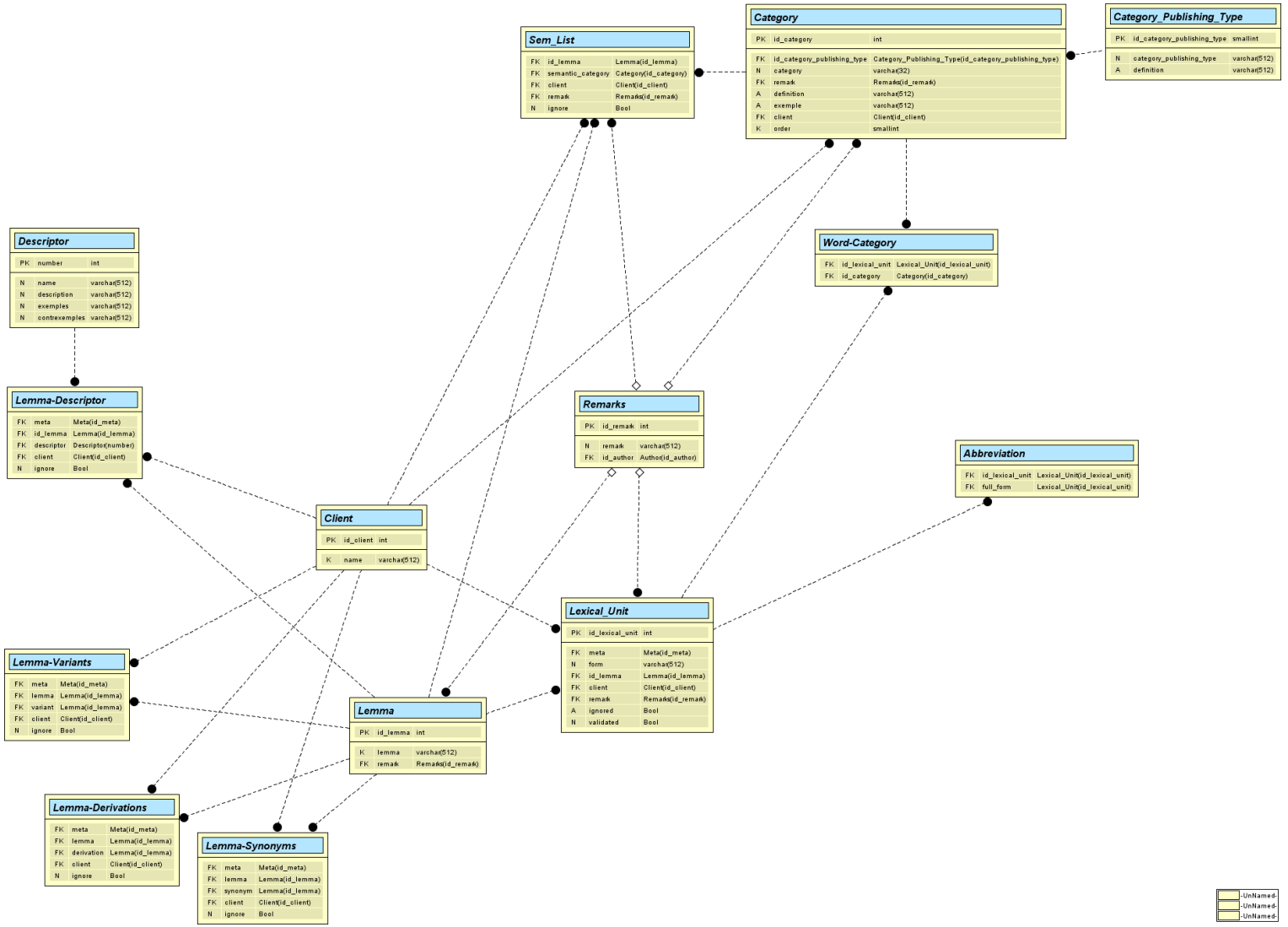


Figure 86 : Schéma relationnel 3.0 « minimal »

Annexe E. Copies d'écran supplémentaires du prototype

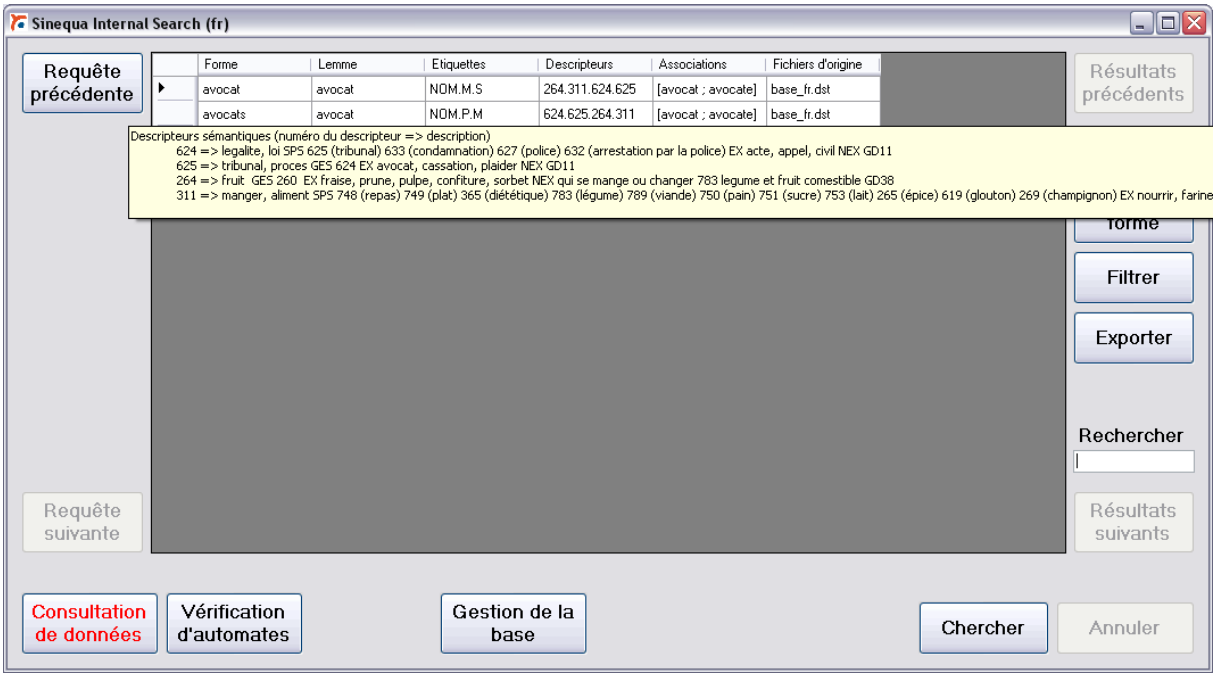


Figure 87 : Fenêtre de résultats du filtre : *Graphie du lemme / est du type / avocat*

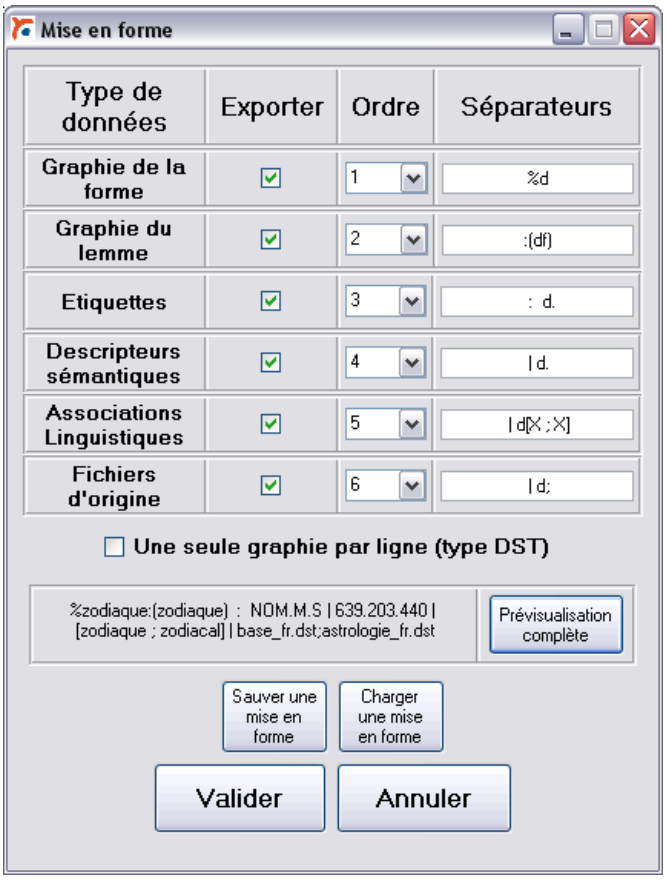


Figure 88 : Fenêtre d'édition de la mise en forme pour l'export de données.

Annexe F. Copies d'écran de l'éditeur de lexiques

L'outil donnant un accès unique et globale à tous les lexiques a été industrialisé par l'équipe de Sinequa[123]. L'accent a été mis sur la consultation dans une première étape, la modification étant remise à une étape ultérieure. La figure 89 est une copie d'écran de cette interface après chargement des dictionnaires de l'anglais. Lors du chargement, un nombre élevé de tests de cohérence est effectué pour vérifier la correspondance entre les données et l'architecture linguistique.

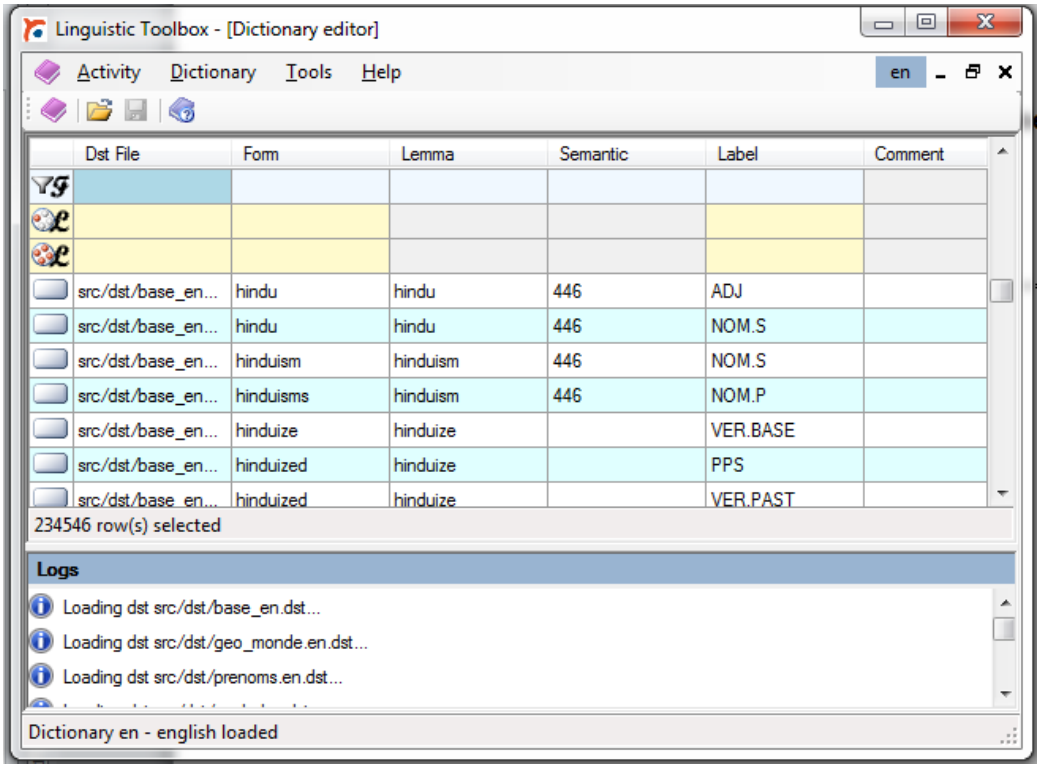


Figure 89 : Interface de l'éditeur de dictionnaires

Les informations agrégées sont présentées dans des colonnes. La première, *Dst file*, contient l'origine des informations, c'est-à-dire le nom du fichier contenant l'entrée lexicale. Ensuite suivent les colonnes avec les mots-forme, *Form*, avec les lemmes, *Lemma*, les codes sémantiques associés aux lemmes, *Semantic*, et les catégories grammaticales, *Label*. Dans la dernière colonne sont affichés les commentaires s'il y en a.

Cet outil permet un filtrage multicritères sur les colonnes dans les trois premières lignes affichées. La figure 90 donne un exemple de filtrage simple sur le mot *hindu*, affichant l'adjectif, puis le nom au singulier et au pluriel.

Le langage de filtrage est évolué : on bénéficie de la puissance des expressions régulières pour les critères des mots-formes et des lemmes, et les catégories grammaticales et sémantiques peuvent être saisies avec combinaisons logiques. Par exemple la requête NOM&(VER|ADJ)

[123] Plus particulièrement par Henri-Emmanuel Ribeiro, qui a entièrement revu l'ergonomie avec l'aide des utilisateurs.

dans *Label* cherche tous les mots qui sont nom et verbe ou adjectif, donc tous ceux qui sont nom et verbe, nom et adjectif, nom et verbe et adjectif.

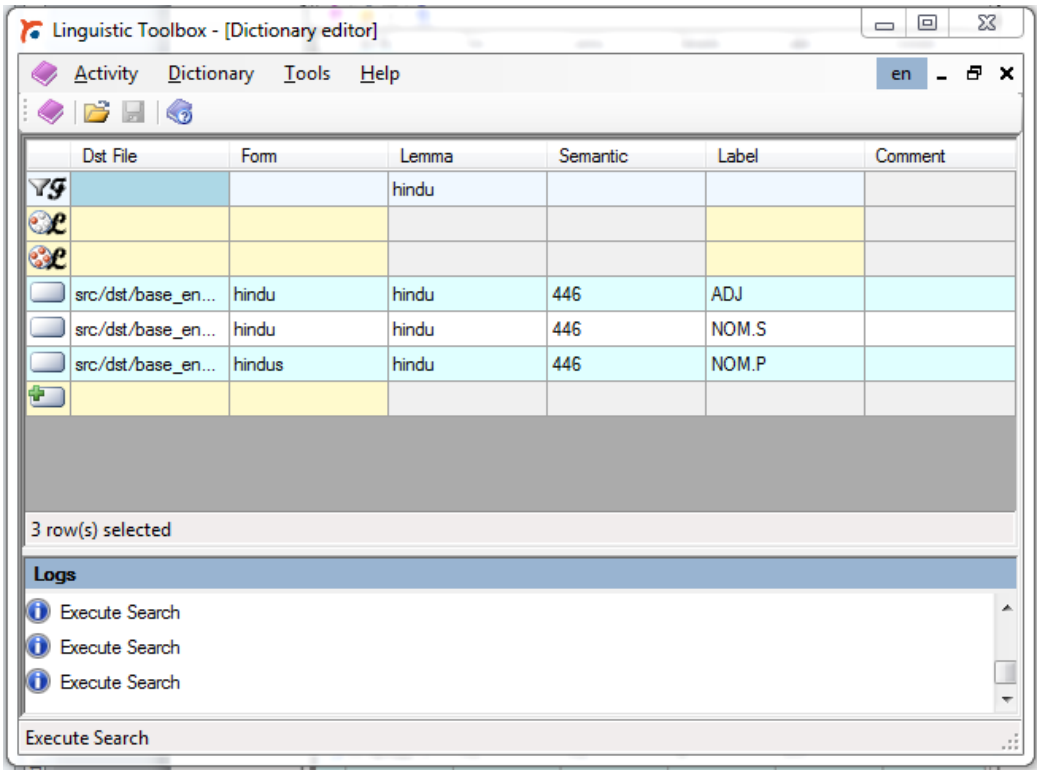


Figure 90: Interface de l'éditeur de dictionnaires (filtrage sur le lemme *hindu*)

Dans la deuxième et troisième ligne, marquées par *L*, on sélectionne tous les mots-formes associés aux lemmes si les mots- correspondent aux critères de filtrage. Ces critères portent sur le mot-forme et sur les catégories grammaticales.

Pour la deuxième ligne, il suffit qu'un seul mot-forme dans les mots-formes associés au lemme corresponde aux critères pour que tous les mots-formes associés à ces lemmes s'affichent. Pour la troisième ligne, tous les mots-formes associés au même lemme doivent correspondre aux critères.

L'implémentation des autres fonctionnalités, notamment la vérification des grammaires et l'accès en écriture, est prévue dans un avenir proche.

Annexe G. Exemples de bonnes hypothèses avec les seuils à 500 et 10

Prés. Rac.	Racine	Terminaison	Score d'entropie	Description
1	therapieresiste	nter	1.74532261375115	[ADJ.COMP.0-89][ADJ.ER-122][NOM.A.M.P-19][NOM.A.M.S-20][NOM.D.M.S-20][NOM.G.M.P-27][NOM.N.M.P-19] [NOM.N.M.S-28][VER.PPS.ER-223]
1	frauenfeind	lich	0.620187444621208	[ADJ.0-468][VER.II.IL-27][VER.II.JE-27][VER.IM.TU-14][VER.PI.JE-14]
1	unspezifiz	iert	1.44570647062607	[ADJ.0-54][VER.IM.VOUS-841][VER.PI.IL-842][VER.PI.VOUS-841][VER.PPS.0-830]
1	unregul	iert	1.44570647062607	[ADJ.0-54][VER.IM.VOUS-841][VER.PI.IL-842][VER.PI.VOUS-841][VER.PPS.0-830]
1	chromosoma	ler	2.13616191146089	[ADJ.COMP.0-343][ADJ.ER-482][NOM.A.M.P-355][NOM.A.M.S-359][NOM.D.M.S-359][NOM.G.F.P-17] [NOM.G.M.P-372][NOM.N.M.P-355][NOM.N.M.S-377][VER.IM.TU-14][VER.PI.JE-14]
1	elfmona	tigen	2.09289209271402	[ADJ.EN-214][NOM.A.M.P-13][NOM.A.M.S-13][NOM.D.M.P-13][NOM.D.M.S-13][NOM.G.M.P-13][NOM.G.M.S-13] [NOM.N.M.P-13][VER.IN-61][VER.PI.ILS-61][VER.PI.NOUS-61][VER.SUP.ILS-61][VER.SUP.NOUS-61]
1	mitzulief	ern	1.98155876486635	[ADJ.0-18][NOM.A.F.P-72][NOM.D.F.P-75][NOM.D.M.P-2657][NOM.D.M.S-11][NOM.D.MX.P-44][NOM.D.X.P-239] [NOM.G.F.P-72][NOM.N.F.P-72][VER.IM.TU-16][VER.IN-582][VER.PI.ILS-582][VER.PI.JE-16][VER.PI.NOUS-582] [VER.SUP.ILS-582][VER.SUP.NOUS-582][VER.ZU-202]
1	heruntertransform	iert	1.44570647062607	[ADJ.0-54][VER.IM.VOUS-841][VER.PI.IL-842][VER.PI.VOUS-841][VER.PPS.0-830]
1	Programmierbar	keit	1.38629436111989	[NOM.A.F.S-1371][NOM.D.F.S-1371][NOM.G.F.S-1371][NOM.N.F.S-1371]
1	Konsensustheo	rie	1.38629436111989	[NOM.A.F.S-217][NOM.D.F.S-217][NOM.G.F.S-217][NOM.N.F.S-217]
1	Interpretierbar	keit	1.38629436111989	[NOM.A.F.S-1371][NOM.D.F.S-1371][NOM.G.F.S-1371][NOM.N.F.S-1371]
Prés. Rac.	Racine	Terminaison	Score d'entropie	Description
2	nichtfiktion	alen	2.43225887195374	[ADJ.EN-259][NOM.A.F.P-40][NOM.A.M.P-24][NOM.A.M.S-23][NOM.D.F.P-43][NOM.D.M.P-68][NOM.D.M.S-23] [NOM.D.X.P-69][NOM.G.F.P-40][NOM.G.M.P-24][NOM.G.M.S-23][NOM.N.F.P-40][NOM.N.M.P-24][VER.IN-15] [VER.PI.ILS-15][VER.PI.NOUS-15][VER.SUP.ILS-15][VER.SUP.NOUS-15]
2	uninitiali	sierte	1.6094379124341	[VER.II.IL-160][VER.II.JE-160][VER.PPS.E-160][VER.SUI.IL-160][VER.SUI.JE-160]
2	uninitiali	sierten	1.6094379124341	[VER.II.ILS-160][VER.II.NOUS-160][VER.PPS.EN-160][VER.SUI.ILS-160][VER.SUI.NOUS-160]
2	unpolarisi	ertem	0.228135236147046	[ADJ.EM-91][VER.PPS.EM-1415]
2	unpolarisi	ertes	0.228135236147046	[ADJ.ES-91][VER.PPS.ES-1415]
2	karamelisi	ertem	0.228135236147046	[ADJ.EM-91][VER.PPS.EM-1415]
2	differenzierba	rer	1.2074244507603	[ADJ.COMP.0-251][ADJ.COMP.ER-3891][ADJ.ER-382][NOM.A.M.P-165][NOM.A.M.S-165][NOM.D.M.S-165] [NOM.G.M.P-172][NOM.N.M.P-165][NOM.N.M.S-172]
2	differenzierba	res	0.827993923437492	[ADJ.COMP.ES-3891][ADJ.ES-378][NOM.A.M.P-17][NOM.A.X.P-13][NOM.D.M.P-17][NOM.D.X.P-13][NOM.G.M.P-17] [NOM.G.M.S-377][NOM.G.X.P-13][NOM.G.X.S-152][NOM.N.M.P-17][NOM.N.X.P-13]
2	nicht-fiktion	alen	2.43225887195374	[ADJ.EN-259][NOM.A.F.P-40][NOM.A.M.P-24][NOM.A.M.S-23][NOM.D.F.P-43][NOM.D.M.P-68][NOM.D.M.S-23] [NOM.D.X.P-69][NOM.G.F.P-40][NOM.G.M.P-24][NOM.G.M.S-23][NOM.N.F.P-40][NOM.N.M.P-24][VER.IN-15] [VER.PI.ILS-15][VER.PI.NOUS-15][VER.SUP.ILS-15][VER.SUP.NOUS-15]
2	hochartifizi	elle	2.45275689185127	[ADJ.E-81][NOM.A.F.S-64][NOM.A.M.P-14][NOM.A.X.P-19][NOM.D.F.S-59][NOM.G.F.S-59][NOM.G.X.P-19] [NOM.N.F.S-64][NOM.N.M.P-14][NOM.N.X.P-19][VER.IM.TU-95][VER.PI.JE-97][VER.SUP.IL-97][VER.SUP.JE-97]
2	hochartifizi	ellen	2.4832396631456	[ADJ.EN-81][NOM.A.F.P-68][NOM.D.F.P-68][NOM.D.M.P-16][NOM.D.X.P-21][NOM.G.F.P-68][NOM.N.F.P-68] [VER.IN-97][VER.PI.ILS-97][VER.PI.NOUS-97][VER.SUP.ILS-97][VER.SUP.NOUS-97][VER.ZU-74]
2	DDR-Nationalspie	ler	2.13616191146089	[ADJ.COMP.0-343][ADJ.ER-482][NOM.A.M.P-355][NOM.A.M.S-359][NOM.D.M.S-359][NOM.G.F.P-17] [NOM.G.M.P-372][NOM.N.M.P-355][NOM.N.M.S-377][VER.IM.TU-14][VER.PI.JE-14]

2	Verflac	hung	1.38629436111989	[NOM.A.F.S-139][NOM.D.F.S-139][NOM.G.F.S-139][NOM.N.F.S-139]
2	Zellkompartimen	te	2.48402485664298	[ADJ.E-692][ADJ.SUPER.E-4137][NOM.A.F.P-163][NOM.A.F.S-566][NOM.A.M.P-529][NOM.A.X.P-512][NOM.A.X.S-29] [NOM.D.F.S-464][NOM.D.X.S-21][NOM.G.F.P-60][NOM.G.F.S-468][NOM.G.M.P-417][NOM.G.X.P-511] [NOM.N.F.P-163][NOM.N.F.S-566][NOM.N.M.P-529][NOM.N.M.S-168][NOM.N.X.P-512][NOM.N.X.S-29] [VER.II.IL-6756][VER.II.JE-6751][VER.IM.TU-695][VER.PI.JE-727][VER.PPS.E-6684][VER.SUI.IL-7002] [VER.SUI.JE-6995][VER.SUP.IL-728][VER.SUP.JE-727]
2	Bagatellisier	ung	1.63716246998186	[NOM.A.F.S-2220][NOM.A.M.S-169][NOM.D.F.S-2221][NOM.D.M.S-169][NOM.G.F.S-2220][NOM.G.M.S-152] [NOM.N.F.S-2223][NOM.N.M.S-168]
2	Bruttoinlandprodu	kt	1.82941469266949	[ADJ.0-32][NOM.A.M.S-42][NOM.A.X.S-28][NOM.D.M.S-42][NOM.D.X.S-28][NOM.N.M.S-46][NOM.N.X.S-28] [VER.II.VOUS-26][VER.IM.VOUS-513][VER.PI.IL-524][VER.PI.VOUS-512][VER.PPS.0-470]
2	Bruttoinlandprodu	ktes	0.645966021261465	[ADJ.ES-32][NOM.G.M.S-41][NOM.G.X.S-25][VER.PPS.ES-470]
2	Familiennna	me	2.72865809919065	[ADJ.E-84][NOM.A.F.S-122][NOM.A.M.P-107][NOM.A.X.P-203][NOM.D.F.S-120][NOM.G.F.S-120][NOM.G.M.P-104] [NOM.G.X.P-203][NOM.N.F.S-122][NOM.N.M.P-107][NOM.N.M.S-25][NOM.N.X.P-203][VER.IM.TU-334] [VER.PI.JE-382][VER.SUI.IL-187][VER.SUI.JE-187][VER.SUP.IL-382][VER.SUP.JE-382]

Prés. Rac.	Racine	Terminaison	Score d'entropie	Description
3	steckb	ar	2.12927191450529	[ADJ.0-209][NOM.A.M.S-91][NOM.A.X.S-54][NOM.D.M.S-91][NOM.D.X.S-54][NOM.N.M.S-101][NOM.N.X.S-54] [VER.II.IL-30][VER.II.JE-30][VER.IM.TU-15][VER.PI.JE-15]
3	steckb	are	2.26072559948437	[ADJ.E-209][NOM.A.F.S-13][NOM.A.M.P-77][NOM.A.X.P-39][NOM.D.F.S-13][NOM.G.F.S-13][NOM.G.M.P-77] [NOM.G.X.P-39][NOM.N.F.S-13][NOM.N.M.P-77][NOM.N.X.P-39][VER.IM.TU-15][VER.PI.JE-16][VER.SUP.IL-16] [VER.SUP.JE-16]
3	steckb	aren	2.49800933655241	[ADJ.EN-209][NOM.A.F.P-16][NOM.A.M.P-19][NOM.A.M.S-16][NOM.D.F.P-16][NOM.D.M.P-95][NOM.D.M.S-16] [NOM.D.X.P-39][NOM.G.F.P-16][NOM.G.M.P-19][NOM.G.M.S-16][NOM.N.F.P-16][NOM.N.M.P-19][VER.II.ILS-30] [VER.II.NOUS-30][VER.IN-16][VER.PI.ILS-16][VER.PI.NOUS-16][VER.SUP.ILS-16][VER.SUP.NOUS-16]
3	feldornitholog	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
3	hyperre	elle	2.45275689185127	[ADJ.E-81][NOM.A.F.S-64][NOM.A.M.P-14][NOM.A.X.P-19][NOM.D.F.S-59][NOM.G.F.S-59][NOM.G.X.P-19] [NOM.N.F.S-64][NOM.N.M.P-14][NOM.N.X.P-19][VER.IM.TU-95][VER.PI.JE-97][VER.SUP.IL-97][VER.SUP.JE-97]
3	hyperre	ellen	2.4832396631456	[ADJ.EN-81][NOM.A.F.P-68][NOM.D.F.P-68][NOM.D.M.P-16][NOM.D.X.P-21][NOM.G.F.P-68][NOM.N.F.P-68] [VER.IN-97][VER.PI.ILS-97][VER.PI.NOUS-97][VER.SUP.ILS-97][VER.SUP.NOUS-97][VER.ZU-74]
3	zuwiderlie	f	2.10058877313632	[ADJ.0-15][NOM.A.M.S-177][NOM.A.X.S-41][NOM.D.M.S-177][NOM.D.X.S-41][NOM.N.M.S-249][NOM.N.X.S-41] [VER.II.IL-214][VER.II.JE-214][VER.IM.TU-428][VER.PI.JE-381]
3	zuwiderlie	fen	2.94150184363736	[ADJ.0-14][ADJ.EN-15][NOM.A.F.P-62][NOM.A.M.P-106][NOM.A.M.S-105][NOM.D.F.P-63][NOM.D.M.P-247] [NOM.D.M.S-105][NOM.D.X.P-25][NOM.G.F.P-62][NOM.G.M.P-106][NOM.G.M.S-83][NOM.N.F.P-62][NOM.N.M.P-106] [NOM.N.M.S-23][VER.II.ILS-214][VER.II.NOUS-214][VER.IN-433][VER.PI.ILS-434][VER.PI.NOUS-434] [VER.PPS.0-214][VER.SUI.ILS-275][VER.SUI.NOUS-275][VER.SUP.ILS-434][VER.SUP.NOUS-434][VER.ZU-287]
3	unlimi	tiert	1.38628919157507	[VER.IM.VOUS-134][VER.PI.IL-135][VER.PI.VOUS-135][VER.PPS.0-135]
3	unlimi	tierte	1.6094379124341	[VER.II.IL-135][VER.II.JE-135][VER.PPS.E-135][VER.SUI.IL-135][VER.SUI.JE-135]
3	randomisi	erter	0.603150906406462	[ADJ.COMP.0-76][ADJ.ER-91][NOM.G.F.P-19][NOM.G.M.P-23][NOM.N.M.S-23][VER.PPS.ER-1415]
3	Metallur	ge	2.71987092142039	[ADJ.E-772][NOM.A.F.P-52][NOM.A.F.S-296][NOM.A.M.P-422][NOM.A.X.P-35][NOM.A.X.S-38][NOM.D.F.S-259] [NOM.D.X.S-34][NOM.G.F.P-15][NOM.G.F.S-260][NOM.G.M.P-374][NOM.G.X.P-35][NOM.N.F.P-52][NOM.N.F.S-296] [NOM.N.M.P-422][NOM.N.M.S-185][NOM.N.X.P-35][NOM.N.X.S-38][VER.IM.TU-1013][VER.PI.JE-954][VER.SUI.IL-515] [VER.SUI.JE-515][VER.SUP.IL-956][VER.SUP.JE-956]
3	Aufle	gung	1.38629436111989	[NOM.A.F.S-222][NOM.D.F.S-222][NOM.G.F.S-222][NOM.N.F.S-222]
3	Vorstud	ie	1.54922384295844	[ADJ.E-17][NOM.A.F.S-1525][NOM.A.M.P-14][NOM.A.M.S-12][NOM.D.F.S-1524][NOM.D.M.S-12][NOM.G.F.S-1524] [NOM.G.M.P-13][NOM.N.F.S-1525][NOM.N.M.P-14][NOM.N.M.S-14][VER.IM.TU-21][VER.PI.JE-23][VER.SUP.IL-24] [VER.SUP.JE-23]
3	Vorstud	ien	2.00515936981159	[ADJ.EN-19][NOM.A.F.P-1987][NOM.A.M.P-18][NOM.A.X.P-364][NOM.A.X.S-11][NOM.D.F.P-1989][NOM.D.M.P-29]

				[NOM.D.X.P-366][NOM.D.X.S-11][NOM.G.F.P-1987][NOM.G.F.S-44][NOM.G.M.P-17][NOM.G.X.P-364][NOM.N.F.P-1987] [NOM.N.M.P-18][NOM.N.X.P-364][NOM.N.X.S-11][VER.IN-24][VER.PI.ILS-23][VER.PI.NOUS-23][VER.SUP.ILS-53] [VER.SUP.NOUS-53][VER.ZU-12]
3	Vorstud	ium	1.09861228866811	[NOM.A.X.S-376][NOM.D.X.S-376][NOM.N.X.S-376]
3	Befahr	ung	1.63716246998186	[NOM.A.F.S-2220][NOM.A.M.S-169][NOM.D.F.S-2221][NOM.D.M.S-169][NOM.G.F.S-2220][NOM.G.M.S-152][NOM.N.F.S-2223] [NOM.N.M.S-168]
3	Nebenfor	m	2.01365225660997	[ADJ.0-84][ADJ.COMP.EM-3905][ADJ.EM-5075][ADJ.SUPER.EM-4137][NOM.A.F.S-20][NOM.A.M.S-160][NOM.A.X.S-1013] [NOM.D.F.S-20][NOM.D.M.S-161][NOM.D.X.S-1057][NOM.G.F.S-21][NOM.N.F.S-20][NOM.N.M.S-167][NOM.N.X.S-1013] [VER.II.IL-174][VER.II.JE-174][VER.IM.TU-370][VER.PI.JE-367][VER.PPR.EM-8989][VER.PPS.EM-9071]
3	Nebenfor	men	3.19726658299568	[ADJ.0-11][ADJ.EN-84][NOM.A.F.P-142][NOM.A.M.P-212][NOM.A.M.S-44][NOM.A.X.P-55][NOM.A.X.S-49][NOM.D.F.P-142] [NOM.D.M.P-330][NOM.D.M.S-44][NOM.D.X.P-258][NOM.D.X.S-49][NOM.G.F.P-142][NOM.G.M.P-212][NOM.G.M.S-20] [NOM.G.X.P-55][NOM.N.F.P-142][NOM.N.M.P-211][NOM.N.M.S-22][NOM.N.X.P-55][NOM.N.X.S-49][VER.II.ILS-174] [VER.II.NOUS-174][VER.IN-382][VER.PI.ILS-382][VER.PI.NOUS-382][VER.PPS.0-173][VER.SUI.ILS-187][VER.SUI.NOUS-187] [VER.SUP.ILS-382][VER.SUP.NOUS-382][VER.ZU-254]
Prés. Rac.	Racine	Terminaison	Score d'entropie	Description
4	tragikom	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
4	tragikom	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
4	tragikom	isches	0.0646779394943541	[ADJ.ES-1077][NOM.G.M.S-13]
4	exeku	tieren	1.66933668633677	[VER.IN-135][VER.PI.ILS-135][VER.PI.NOUS-135][VER.SUP.ILS-135][VER.SUP.NOUS-135][VER.ZU-12]
4	exeku	tiert	1.38628919157507	[VER.IM.VOUS-134][VER.PI.IL-135][VER.PI.VOUS-135][VER.PPS.0-135]
4	exeku	tierte	1.6094379124341	[VER.II.IL-135][VER.II.JE-135][VER.PPS.E-135][VER.SUI.IL-135][VER.SUI.JE-135]
4	exeku	tierten	1.6094379124341	[VER.II.ILS-135][VER.II.NOUS-135][VER.PPS.EN-135][VER.SUI.ILS-135][VER.SUI.NOUS-135]
4	lokalisierb	ar	2.12927191450529	[ADJ.0-209][NOM.A.M.S-91][NOM.A.X.S-54][NOM.D.M.S-91][NOM.D.X.S-54][NOM.N.M.S-101][NOM.N.X.S-54][VER.II.IL-30] [VER.II.JE-30][VER.IM.TU-15][VER.PI.JE-15]
4	lokalisierb	are	2.26072559948437	[ADJ.E-209][NOM.A.F.S-13][NOM.A.M.P-77][NOM.A.X.P-39][NOM.D.F.S-13][NOM.G.F.S-13][NOM.G.M.P-77][NOM.G.X.P-39] [NOM.N.F.S-13][NOM.N.M.P-77][NOM.N.X.P-39][VER.IM.TU-15][VER.PI.JE-16][VER.SUP.IL-16][VER.SUP.JE-16]
4	lokalisierb	aren	2.49800933655241	[ADJ.EN-209][NOM.A.F.P-16][NOM.A.M.P-19][NOM.A.M.S-16][NOM.D.F.P-16][NOM.D.M.P-95][NOM.D.M.S-16][NOM.D.X.P-39] [NOM.G.F.P-16][NOM.G.M.P-19][NOM.G.M.S-16][NOM.N.F.P-16][NOM.N.M.P-19][VER.II.ILS-30][VER.II.NOUS-30][VER.IN-16] [VER.PI.ILS-16][VER.PI.NOUS-16][VER.SUP.ILS-16][VER.SUP.NOUS-16]
4	mitot	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
4	mitot	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
4	mitot	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
4	vulkanolog	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
4	wiederauferstand	ene	0.966137771583489	[ADJ.E-189][NOM.A.F.P-31][NOM.A.F.S-50][NOM.A.M.P-40][NOM.A.X.P-28][NOM.D.F.S-21][NOM.G.F.S-22][NOM.G.X.P-28] [NOM.N.F.P-31][NOM.N.F.S-50][NOM.N.M.P-40][NOM.N.M.S-38][NOM.N.X.P-28][VER.PPS.E-2358]
4	wiederauferstand	enen	1.1643626324349	[ADJ.EN-189][NOM.A.F.P-51][NOM.A.M.P-38][NOM.A.M.S-38][NOM.D.F.P-51][NOM.D.F.S-29][NOM.D.M.P-45][NOM.D.M.S-38] [NOM.D.X.P-29][NOM.G.F.P-51][NOM.G.F.S-29][NOM.G.M.P-38][NOM.G.M.S-38][NOM.N.F.P-51][NOM.N.M.P-38] [VER.PPS.EN-2359]
4	zweisitz	ig	1.00053136036751	[ADJ.0-746][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-171][VER.PI.JE-172]
4	zweisitz	ige	2.02535584020433	[ADJ.E-748][NOM.A.F.P-37][NOM.A.F.S-52][NOM.A.M.P-55][NOM.D.F.S-15][NOM.G.F.S-15][NOM.N.F.P-37][NOM.N.F.S-52] [NOM.N.M.P-55][NOM.N.M.S-47][VER.IM.TU-194][VER.PI.JE-187][VER.SUP.IL-187][VER.SUP.JE-187]
4	zweisitz	igen	2.50197765093382	[ADJ.EN-748][NOM.A.F.P-62][NOM.A.M.P-48][NOM.A.M.S-48][NOM.D.F.P-62][NOM.D.F.S-37][NOM.D.M.P-56][NOM.D.M.S-48] [NOM.G.F.P-62][NOM.G.F.S-38][NOM.G.M.P-48][NOM.G.M.S-47][NOM.N.F.P-62][NOM.N.M.P-48][VER.IN-187][VER.PI.ILS-187] [VER.PI.NOUS-187][VER.SUP.ILS-187][VER.SUP.NOUS-187][VER.ZU-49]
4	zweisitz	iges	0	[ADJ.ES-746]

4	Sappe	ur	2.10027720202547	[NOM.A.F.S-149][NOM.A.M.S-148][NOM.A.X.S-18][NOM.D.F.S-149][NOM.D.M.S-148][NOM.D.X.S-18][NOM.G.F.S-148] [NOM.N.F.S-149][NOM.N.M.S-153][NOM.N.X.S-18]
4	Sappe	ure	2.09055271782333	[NOM.A.F.S-15][NOM.A.M.P-132][NOM.D.F.S-15][NOM.G.F.S-15][NOM.G.M.P-131][NOM.N.F.S-15][NOM.N.M.P-132] [VER.IM.TU-48][VER.PI.JE-47][VER.SUP.IL-47][VER.SUP.JE-47]
4	Sappe	uren	1.76311238115483	[NOM.A.F.P-164][NOM.A.M.P-12][NOM.D.F.P-166][NOM.D.M.P-143][NOM.G.F.P-165][NOM.G.M.P-12][NOM.N.F.P-165] [NOM.N.M.P-12]
4	Physiokrat	ie	1.54922384295844	[ADJ.E-17][NOM.A.F.S-1525][NOM.A.M.P-14][NOM.A.M.S-12][NOM.D.F.S-1524][NOM.D.M.S-12][NOM.G.F.S-1524][NOM.G.M.P-13] [NOM.N.F.S-1525][NOM.N.M.P-14][NOM.N.M.S-14][VER.IM.TU-21][VER.PI.JE-23][VER.SUP.IL-24][VER.SUP.JE-23]
4	Organiz	ismus	1.38629401396632	[NOM.A.M.S-520][NOM.D.M.S-520][NOM.G.M.S-519][NOM.N.M.S-520]
4	Agrarflugzeu	g	2.40635141202603	[ADJ.0-766][NOM.A.F.S-2252][NOM.A.M.S-581][NOM.A.X.S-99][NOM.D.F.S-2253][NOM.D.M.S-581][NOM.D.X.S-99] [NOM.G.F.S-2252][NOM.G.M.S-157][NOM.N.F.S-2255][NOM.N.M.S-586][NOM.N.X.S-99][VER.II.IL-515][VER.II.JE-515] [VER.IM.TU-986][VER.PI.JE-930]
4	Agrarflugzeu	ge	2.71987092142039	[ADJ.E-772][NOM.A.F.P-52][NOM.A.F.S-296][NOM.A.M.P-422][NOM.A.X.P-35][NOM.A.X.S-38][NOM.D.F.S-259][NOM.D.X.S-34] [NOM.G.F.P-15][NOM.G.F.S-260][NOM.G.M.P-374][NOM.G.X.P-35][NOM.N.F.P-52][NOM.N.F.S-296][NOM.N.M.P-422] [NOM.N.M.S-185][NOM.N.X.P-35][NOM.N.X.S-38][VER.IM.TU-1013][VER.PI.JE-954][VER.SUI.IL-515][VER.SUI.JE-515] [VER.SUP.IL-956][VER.SUP.JE-956]
4	Anhaft	ung	1.63716246998186	[NOM.A.F.S-2220][NOM.A.M.S-169][NOM.D.F.S-2221][NOM.D.M.S-169][NOM.G.F.S-2220][NOM.G.M.S-152][NOM.N.F.S-2223] [NOM.N.M.S-168]
4	Anhaft	ungen	1.66649352118057	[NOM.A.F.P-2159][NOM.A.M.P-152][NOM.D.F.P-2159][NOM.D.M.P-159][NOM.G.F.P-2159][NOM.G.M.P-151][NOM.N.F.P-2161] [NOM.N.M.P-151][VER.PPS.0-83]
4	Wiedereintrit	ts	1.96510055004588	[NOM.A.M.P-101][NOM.A.X.P-191][NOM.D.M.P-101][NOM.D.X.P-191][NOM.G.M.P-101][NOM.G.M.S-632][NOM.G.X.P-191] [NOM.G.X.S-843][NOM.N.M.P-101][NOM.N.X.P-191]
4	Hollywoodfil	m	2.01365225660997	[ADJ.0-84][ADJ.COMP.EM-3905][ADJ.EM-5075][ADJ.SUPER.EM-4137][NOM.A.F.S-20][NOM.A.M.S-160][NOM.A.X.S-1013] [NOM.D.F.S-20][NOM.D.M.S-161][NOM.D.X.S-1057][NOM.G.F.S-21][NOM.N.F.S-20][NOM.N.M.S-167][NOM.N.X.S-1013] [VER.II.IL-174][VER.II.JE-174][VER.IM.TU-370][VER.PI.JE-367][VER.PPR.EM-8989][VER.PPS.EM-9071]
4	Hollywoodfil	me	2.72865809919065	[ADJ.E-84][NOM.A.F.S-122][NOM.A.M.P-107][NOM.A.X.P-203][NOM.D.F.S-120][NOM.G.F.S-120][NOM.G.M.P-104] [NOM.G.X.P-203][NOM.N.F.S-122][NOM.N.M.P-107][NOM.N.M.S-25][NOM.N.X.P-203][VER.IM.TU-334][VER.PI.JE-382] [VER.SUI.IL-187][VER.SUI.JE-187][VER.SUP.IL-382][VER.SUP.JE-382]
4	Hollywoodfil	men	3.19726658299568	[ADJ.0-11][ADJ.EN-84][NOM.A.F.P-142][NOM.A.M.P-212][NOM.A.M.S-44][NOM.A.X.P-55][NOM.A.X.S-49][NOM.D.F.P-142] [NOM.D.M.P-330][NOM.D.M.S-44][NOM.D.X.P-258][NOM.D.X.S-49][NOM.G.F.P-142][NOM.G.M.P-212][NOM.G.M.S-20] [NOM.G.X.P-55][NOM.N.F.P-142][NOM.N.M.P-211][NOM.N.M.S-22][NOM.N.X.P-55][NOM.N.X.S-49][VER.II.ILS-174] [VER.II.NOUS-174][VER.IN-382][VER.PI.ILS-382][VER.PI.NOUS-382][VER.PPS.0-173][VER.SUI.ILS-187][VER.SUI.NOUS-187] [VER.SUP.ILS-382][VER.SUP.NOUS-382][VER.ZU-254]
4	Hollywoodfil	ms	1.07887954450763	[NOM.A.M.P-22][NOM.A.X.P-31][NOM.D.M.P-22][NOM.D.X.P-31][NOM.G.M.P-22][NOM.G.M.S-156][NOM.G.X.P-31] [NOM.G.X.S-1012][NOM.N.M.P-22][NOM.N.X.P-31]
4	Nebeneffe	kt	1.82941469266949	[ADJ.0-32][NOM.A.M.S-42][NOM.A.X.S-28][NOM.D.M.S-42][NOM.D.X.S-28][NOM.N.M.S-46][NOM.N.X.S-28][VER.II.VOUS-26] [VER.IM.VOUS-513][VER.PI.IL-524][VER.PI.VOUS-512][VER.PPS.0-470]
4	Nebeneffe	kte	1.92680461734962	[ADJ.E-32][NOM.A.M.P-39][NOM.A.X.P-25][NOM.G.M.P-37][NOM.G.X.P-25][NOM.N.M.P-39][NOM.N.X.P-25][VER.II.IL-481] [VER.II.JE-481][VER.PPS.E-470][VER.SUI.IL-481][VER.SUI.JE-481]
4	Nebeneffe	kten	1.76115627789193	[ADJ.EN-32][NOM.D.M.P-46][NOM.D.X.P-27][VER.II.ILS-481][VER.II.NOUS-481][VER.PPS.EN-470][VER.SUI.ILS-481] [VER.SUI.NOUS-481]
4	Aranes	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
4	Aranes	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
4	Zipf	sche	1.93530962289556	[ADJ.E-1092][NOM.A.F.S-88][NOM.A.M.P-63][NOM.D.F.S-83][NOM.G.F.S-83][NOM.G.M.P-58][NOM.N.F.S-88][NOM.N.M.P-63] [VER.IM.TU-155][VER.PI.JE-160][VER.SUI.IL-12][VER.SUI.JE-12][VER.SUP.IL-160][VER.SUP.JE-160]
4	Zipf	schen	2.25777783102144	[ADJ.EN-1092][NOM.A.F.P-93][NOM.A.M.P-13][NOM.A.M.S-12][NOM.A.X.S-11][NOM.D.F.P-93][NOM.D.M.P-69][NOM.D.M.S-12] [NOM.D.X.P-12][NOM.D.X.S-14][NOM.G.F.P-93][NOM.G.M.P-13][NOM.N.F.P-93][NOM.N.M.P-13][NOM.N.X.S-11][VER.II.ILS-12] [VER.II.NOUS-12][VER.IN-159][VER.PI.ILS-160][VER.PI.NOUS-160][VER.PPS.0-12][VER.SUI.ILS-12][VER.SUI.NOUS-12]

				[VER.SUP.ILS-160][VER.SUP.NOUS-160][VER.ZU-71]
4	Zipf	sches	0.305620250653627	[ADJ.ES-1092][NOM.G.M.S-76][NOM.G.X.S-15]
4	Rhinocero	s	2.69537712595293	[ADJ.0-216][ADJ.COMP.ES-3892][ADJ.ES-5072][ADJ.SUPER.ES-4136][NOM.A.F.P-329][NOM.A.F.S-305][NOM.A.M.P-1055] [NOM.A.M.S-1087][NOM.A.X.P-873][NOM.A.X.S-235][NOM.D.F.P-329][NOM.D.F.S-305][NOM.D.M.P-1049][NOM.D.M.S-1087] [NOM.D.X.P-873][NOM.D.X.S-191][NOM.G.F.P-329][NOM.G.F.S-330][NOM.G.M.P-1055][NOM.G.M.S-11174][NOM.G.MX.S-60] [NOM.G.X.P-873][NOM.G.X.S-7538][NOM.N.F.P-330][NOM.N.F.S-306][NOM.N.M.P-1052][NOM.N.M.S-1089][NOM.N.X.P-873] [NOM.N.X.S-235][VER.II.IL-72][VER.II.JE-72][VER.IM.TU-214][VER.PI.JE-211][VER.PPR.ES-8989][VER.PPS.ES-9076]
4	Klammer	ung	1.63716246998186	[NOM.A.F.S-2220][NOM.A.M.S-169][NOM.D.F.S-2221][NOM.D.M.S-169][NOM.G.F.S-2220][NOM.G.M.S-152][NOM.N.F.S-2223] [NOM.N.M.S-168]
4	Reprotechn	ik	1.84181135452888	[NOM.A.F.P-44][NOM.A.F.S-345][NOM.A.M.S-12][NOM.D.F.P-44][NOM.D.F.S-345][NOM.D.M.S-12][NOM.G.F.P-44][NOM.G.F.S-345] [NOM.N.F.P-44][NOM.N.F.S-345][NOM.N.M.S-17]
Prés. Rac.	Racine	Terminaison	Score d'entropie	Description
5	asymptomat	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
5	asymptomat	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
5	asymptomat	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
5	orchestra	le	2.64423867983132	[ADJ.E-484][NOM.A.F.P-26][NOM.A.F.S-327][NOM.A.M.P-178][NOM.A.M.S-13][NOM.A.X.P-191][NOM.A.X.S-70][NOM.D.F.S-313] [NOM.D.M.S-13][NOM.D.X.S-70][NOM.G.F.S-313][NOM.G.M.P-161][NOM.G.X.P-191][NOM.G.X.S-21][NOM.N.F.P-26] [NOM.N.F.S-327][NOM.N.M.P-178][NOM.N.M.S-58][NOM.N.X.P-191][NOM.N.X.S-71][VER.IM.TU-1155][VER.PI.JE-1126] [VER.SUI.IL-73][VER.SUI.JE-73][VER.SUP.IL-1148][VER.SUP.JE-1146]
5	orchestra	ler	2.13616191146089	[ADJ.COMP.0-343][ADJ.ER-482][NOM.A.M.P-355][NOM.A.M.S-359][NOM.D.M.S-359][NOM.G.F.P-17][NOM.G.M.P-372] [NOM.N.M.P-355][NOM.N.M.S-377][VER.IM.TU-14][VER.PI.JE-14]
5	orchestra	les	1.62346096609502	[ADJ.ES-483][NOM.A.M.P-16][NOM.A.X.P-22][NOM.D.M.P-16][NOM.D.X.P-22][NOM.G.M.P-16][NOM.G.M.S-175][NOM.G.X.P-22] [NOM.G.X.S-272][NOM.N.M.P-16][NOM.N.X.P-22][VER.PI.JE-17]
5	wiedergebor	ene	0.966137771583489	[ADJ.E-189][NOM.A.F.P-31][NOM.A.F.S-50][NOM.A.M.P-40][NOM.A.X.P-28][NOM.D.F.S-21][NOM.G.F.S-22][NOM.G.X.P-28] [NOM.N.F.P-31][NOM.N.F.S-50][NOM.N.M.P-40][NOM.N.M.S-38][NOM.N.X.P-28][VER.PPS.E-2358]
5	wiedergebor	enen	1.1643626324349	[ADJ.EN-189][NOM.A.F.P-51][NOM.A.M.P-38][NOM.A.M.S-38][NOM.D.F.P-51][NOM.D.F.S-29][NOM.D.M.P-45][NOM.D.M.S-38] [NOM.D.X.P-29][NOM.G.F.P-51][NOM.G.F.S-29][NOM.G.M.P-38][NOM.G.M.S-38][NOM.N.F.P-51][NOM.N.M.P-38] [VER.PPS.EN-2359]
5	wiedergebor	ener	0.62531967045454	[ADJ.COMP.0-121][ADJ.ER-189][NOM.G.F.P-29][NOM.G.M.P-39][NOM.N.M.S-39][VER.PPS.ER-2359]
5	ehebrecher	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
5	ehebrecher	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
5	neurolept	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
5	neurolept	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
5	gleichrang	ig	1.00053136036751	[ADJ.0-746][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-171][VER.PI.JE-172]
5	gleichrang	ige	2.02535584020433	[ADJ.E-748][NOM.A.F.P-37][NOM.A.F.S-52][NOM.A.M.P-55][NOM.D.F.S-15][NOM.G.F.S-15][NOM.N.F.P-37][NOM.N.F.S-52] [NOM.N.M.P-55][NOM.N.M.S-47][VER.IM.TU-194][VER.PI.JE-187][VER.SUP.IL-187][VER.SUP.JE-187]
5	gleichrang	igen	2.50197765093382	[ADJ.EN-748][NOM.A.F.P-62][NOM.A.M.P-48][NOM.A.M.S-48][NOM.D.F.P-62][NOM.D.F.S-37][NOM.D.M.P-56][NOM.D.M.S-48] [NOM.G.F.P-62][NOM.G.F.S-38][NOM.G.M.P-48][NOM.G.M.S-47][NOM.N.F.P-62][NOM.N.M.P-48][VER.IN-187][VER.PI.ILS-187] [VER.PI.NOUS-187][VER.SUP.ILS-187][VER.SUP.NOUS-187][VER.ZU-49]
5	gleichrang	iger	1.44402118467646	[ADJ.COMP.0-563][ADJ.ER-747][NOM.A.M.P-36][NOM.A.M.S-36][NOM.D.M.S-36][NOM.G.F.P-38][NOM.G.M.P-83][NOM.N.M.P-36] [NOM.N.M.S-82]
5	gleichrang	iges	0	[ADJ.ES-746]
5	dreigeschoss	ig	1.00053136036751	[ADJ.0-746][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-171][VER.PI.JE-172]
5	dreigeschoss	ige	2.02535584020433	[ADJ.E-748][NOM.A.F.P-37][NOM.A.F.S-52][NOM.A.M.P-55][NOM.D.F.S-15][NOM.G.F.S-15][NOM.N.F.P-37][NOM.N.F.S-52] [NOM.N.M.P-55][NOM.N.M.S-47][VER.IM.TU-194][VER.PI.JE-187][VER.SUP.IL-187][VER.SUP.JE-187]

5	dreigeschoss	igen	2.50197765093382	[ADJ.EN-748][NOM.A.F.P-62][NOM.A.M.P-48][NOM.A.M.S-48][NOM.D.F.P-62][NOM.D.F.S-37][NOM.D.M.P-56][NOM.D.M.S-48] [NOM.G.F.P-62][NOM.G.F.S-38][NOM.G.M.P-48][NOM.G.M.S-47][NOM.N.F.P-62][NOM.N.M.P-48][VER.IN-187][VER.PI.ILS-187] [VER.PI.NOUS-187][VER.SUP.ILS-187][VER.SUP.NOUS-187][VER.ZU-49]
5	dreigeschoss	iger	1.44402118467646	[ADJ.COMP.0-563][ADJ.ER-747][NOM.A.M.P-36][NOM.A.M.S-36][NOM.D.M.S-36][NOM.G.F.P-38][NOM.G.M.P-83][NOM.N.M.P-36] [NOM.N.M.S-82]
5	dreigeschoss	iges	0	[ADJ.ES-746]
5	institutionalisi	erter	0.603150906406462	[ADJ.COMP.0-76][ADJ.ER-91][NOM.G.F.P-19][NOM.G.M.P-23][NOM.N.M.S-23][VER.PPS.ER-1415]
5	institutionalisi	ertes	0.228135236147046	[ADJ.ES-91][VER.PPS.ES-1415]
5	Blogg	erin	1.38629436111989	[NOM.A.F.S-534][NOM.D.F.S-534][NOM.G.F.S-534][NOM.N.F.S-534]
5	Blogg	ern	1.98155876486635	[ADJ.0-18][NOM.A.F.P-72][NOM.D.F.P-75][NOM.D.M.P-2657][NOM.D.M.S-11][NOM.D.MX.P-44][NOM.D.X.P-239][NOM.G.F.P-72] [NOM.N.F.P-72][VER.IM.TU-16][VER.IN-582][VER.PI.ILS-582][VER.PI.JE-16][VER.PI.NOUS-582][VER.SUP.ILS-582] [VER.SUP.NOUS-582][VER.ZU-202]
5	Blogg	ers	0.767836187084297	[NOM.A.M.P-52][NOM.A.X.P-19][NOM.D.M.P-51][NOM.D.X.P-19][NOM.G.M.P-52][NOM.G.M.S-2744][NOM.G.MX.S-45] [NOM.G.X.P-19][NOM.G.X.S-214][NOM.N.M.P-52][NOM.N.X.P-19]
5	Mittelniederdeut	sch	1.39614361053955	[ADJ.0-1092][NOM.A.M.S-86][NOM.A.X.S-20][NOM.D.M.S-86][NOM.D.X.S-19][NOM.N.M.S-88][NOM.N.X.S-20][VER.II.IL-12] [VER.II.JE-12][VER.IM.TU-161][VER.PI.JE-158]
5	Mittelniederdeut	sche	1.93530962289556	[ADJ.E-1092][NOM.A.F.S-88][NOM.A.M.P-63][NOM.D.F.S-83][NOM.G.F.S-83][NOM.G.M.P-58][NOM.N.F.S-88][NOM.N.M.P-63] [VER.IM.TU-155][VER.PI.JE-160][VER.SUI.IL-12][VER.SUI.JE-12][VER.SUP.IL-160][VER.SUP.JE-160]
5	Mittelniederdeut	schen	2.25777783102144	[ADJ.EN-1092][NOM.A.F.P-93][NOM.A.M.P-13][NOM.A.M.S-12][NOM.A.X.S-11][NOM.D.F.P-93][NOM.D.M.P-69][NOM.D.M.S-12] [NOM.D.X.P-12][NOM.D.X.S-14][NOM.G.F.P-93][NOM.G.M.P-13][NOM.N.F.P-93][NOM.N.M.P-13][NOM.N.X.S-11][VER.II.ILS-12] [VER.II.NOUS-12][VER.IN-159][VER.PI.ILS-160][VER.PI.NOUS-160][VER.PPS.0-12][VER.SUI.ILS-12][VER.SUI.NOUS-12] [VER.SUP.ILS-160][VER.SUP.NOUS-160][VER.ZU-71]
5	Mittelniederdeut	sches	0.305620250653627	[ADJ.ES-1092][NOM.G.M.S-76][NOM.G.X.S-15]
5	Nebenzwe	ck	1.74108366437027	[NOM.A.M.S-97][NOM.A.X.S-42][NOM.D.M.S-97][NOM.D.X.S-42][NOM.N.M.S-103][NOM.N.X.S-42][VER.IM.TU-352][VER.PI.JE-341]
5	Nebenzwe	ig	1.00053136036751	[ADJ.0-746][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-171][VER.PI.JE-172]
5	Nebenzwe	ige	2.02535584020433	[ADJ.E-748][NOM.A.F.P-37][NOM.A.F.S-52][NOM.A.M.P-55][NOM.D.F.S-15][NOM.G.F.S-15][NOM.N.F.P-37][NOM.N.F.S-52] [NOM.N.M.P-55][NOM.N.M.S-47][VER.IM.TU-194][VER.PI.JE-187][VER.SUP.IL-187][VER.SUP.JE-187]
5	Herleit	ung	1.63716246998186	[NOM.A.F.S-2220][NOM.A.M.S-169][NOM.D.F.S-2221][NOM.D.M.S-169][NOM.G.F.S-2220][NOM.G.M.S-152][NOM.N.F.S-2223] [NOM.N.M.S-168]
5	Herleit	ungen	1.66649352118057	[NOM.A.F.P-2159][NOM.A.M.P-152][NOM.D.F.P-2159][NOM.D.M.P-159][NOM.G.F.P-2159][NOM.G.M.P-151][NOM.N.F.P-2161] [NOM.N.M.P-151][VER.PPS.0-83]
5	Urspr	ache	1.90024382877499	[ADJ.E-18][NOM.A.F.S-20][NOM.D.F.S-20][NOM.G.F.S-20][NOM.N.F.S-20][VER.IM.TU-112][VER.PI.JE-110][VER.SUP.IL-110] [VER.SUP.JE-110]
5	Urspr	achen	2.39382345978218	[ADJ.EN-18][NOM.A.F.P-19][NOM.D.F.P-19][NOM.D.M.P-13][NOM.G.F.P-19][NOM.N.F.P-19][VER.II.ILS-61][VER.II.NOUS-61] [VER.IN-110][VER.PI.ILS-110][VER.PI.NOUS-110][VER.SUP.ILS-110][VER.SUP.NOUS-110][VER.ZU-87]
5	Anthropolog	in	2.13025375308825	[ADJ.0-20][NOM.A.F.S-836][NOM.A.M.S-88][NOM.A.X.S-277][NOM.D.F.S-836][NOM.D.M.S-88][NOM.D.X.S-277][NOM.G.F.S-838] [NOM.N.F.S-840][NOM.N.M.S-89][NOM.N.X.S-277][VER.IM.TU-16][VER.IN-29][VER.PI.JE-43][VER.ZU-28]
5	Anthropolog	ist	1.70133750601046	[NOM.A.M.S-15][NOM.D.M.S-15][NOM.N.M.S-413][VER.IM.TU-15][VER.IM.VOUS-51][VER.PI.IL-77][VER.PI.JE-15][VER.PI.TU-97] [VER.PI.VOUS-49][VER.PPS.0-28][VER.SUP.TU-29]
Prés. Rac.	Racine	Terminaison	Score d'entropie	Description
6	boole	an	1.88131142591597	[ADJ.0-13][NOM.A.M.S-115][NOM.A.X.S-45][NOM.D.M.S-115][NOM.D.X.S-45][NOM.N.M.S-119][NOM.N.X.S-45][VER.PPS.0-28]
6	boole	sche	1.93530962289556	[ADJ.E-1092][NOM.A.F.S-88][NOM.A.M.P-63][NOM.D.F.S-83][NOM.G.F.S-83][NOM.G.M.P-58][NOM.N.F.S-88][NOM.N.M.P-63] [VER.IM.TU-155][VER.PI.JE-160][VER.SUI.IL-12][VER.SUI.JE-12][VER.SUP.IL-160][VER.SUP.JE-160]
6	boole	schen	2.25777783102144	[ADJ.EN-1092][NOM.A.F.P-93][NOM.A.M.P-13][NOM.A.M.S-12][NOM.A.X.S-11][NOM.D.F.P-93][NOM.D.M.P-69][NOM.D.M.S-12] [NOM.D.X.P-12][NOM.D.X.S-14][NOM.G.F.P-93][NOM.G.M.P-13][NOM.N.F.P-93][NOM.N.M.P-13][NOM.N.X.S-11][VER.II.ILS-12]

				[VER.II.NOUS-12][VER.IN-159][VER.PI.ILS-160][VER.PI.NOUS-160][VER.PPS.0-12][VER.SUI.ILS-12][VER.SUI.NOUS-12] [VER.SUP.ILS-160][VER.SUP.NOUS-160][VER.ZU-71]
6	boole	scher	1.16675840442999	[ADJ.COMP.0-910][ADJ.ER-1095][NOM.A.M.P-40][NOM.A.M.S-42][NOM.D.M.S-42][NOM.G.M.P-45][NOM.N.M.P-40][NOM.N.M.S-47]
6	russifizi	erte	1.77709734163243	[ADJ.E-91][NOM.A.F.P-20][NOM.A.F.S-23][NOM.A.M.P-28][NOM.N.F.P-20][NOM.N.F.S-23][NOM.N.M.P-28][NOM.N.M.S-19] [VER.II.IL-1419][VER.II.JE-1419][VER.IM.TU-11][VER.PPS.E-1415][VER.SUI.IL-1419][VER.SUI.JE-1419]
6	russifizi	erten	1.85574533708063	[ADJ.EN-91][NOM.A.F.P-27][NOM.A.M.P-19][NOM.A.M.S-19][NOM.D.F.P-27][NOM.D.F.S-19][NOM.D.M.P-29][NOM.D.M.S-19] [NOM.G.F.P-27][NOM.G.F.S-20][NOM.G.M.P-19][NOM.G.M.S-19][NOM.N.F.P-27][NOM.N.M.P-19][VER.II.ILS-1419] [VER.II.NOUS-1419][VER.PPS.EN-1415][VER.SUI.ILS-1419][VER.SUI.NOUS-1419]
6	kirgis	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
6	kirgis	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
6	kirgis	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
6	kirgis	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
6	institutionali	sieren	1.6094379124341	[VER.IN-160][VER.PI.ILS-160][VER.PI.NOUS-160][VER.SUP.ILS-160][VER.SUP.NOUS-160]
6	institutionali	siert	1.38629436111989	[VER.IM.VOUS-160][VER.PI.IL-160][VER.PI.VOUS-160][VER.PPS.0-160]
6	institutionali	sierte	1.6094379124341	[VER.II.IL-160][VER.II.JE-160][VER.PPS.E-160][VER.SUI.IL-160][VER.SUI.JE-160]
6	institutionali	sierten	1.6094379124341	[VER.II.ILS-160][VER.II.NOUS-160][VER.PPS.EN-160][VER.SUI.ILS-160][VER.SUI.NOUS-160]
6	burgund	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
6	burgund	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
6	burgund	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
6	burgund	isches	0.0646779394943541	[ADJ.ES-1077][NOM.G.M.S-13]
6	intrigi	eren	1.82872027498886	[ADJ.COMP.EN-3886][ADJ.EN-58][NOM.A.F.P-62][NOM.A.M.P-15][NOM.A.M.S-11][NOM.D.F.P-63][NOM.D.M.P-62] [NOM.D.M.S-11][NOM.D.X.P-33][NOM.G.F.P-62][NOM.G.M.P-15][NOM.G.M.S-11][NOM.N.F.P-62][NOM.N.M.P-15][VER.IN-848] [VER.PI.ILS-849][VER.PI.NOUS-849][VER.SUP.ILS-849][VER.SUP.NOUS-849][VER.ZU-80]
6	intrigi	erte	1.77709734163243	[ADJ.E-91][NOM.A.F.P-20][NOM.A.F.S-23][NOM.A.M.P-28][NOM.N.F.P-20][NOM.N.F.S-23][NOM.N.M.P-28][NOM.N.M.S-19] [VER.II.IL-1419][VER.II.JE-1419][VER.IM.TU-11][VER.PPS.E-1415][VER.SUI.IL-1419][VER.SUI.JE-1419]
6	intrigi	erten	1.85574533708063	[ADJ.EN-91][NOM.A.F.P-27][NOM.A.M.P-19][NOM.A.M.S-19][NOM.D.F.P-27][NOM.D.F.S-19][NOM.D.M.P-29][NOM.D.M.S-19] [NOM.G.F.P-27][NOM.G.F.S-20][NOM.G.M.P-19][NOM.G.M.S-19][NOM.N.F.P-27][NOM.N.M.P-19][VER.II.ILS-1419] [VER.II.NOUS-1419][VER.PPS.EN-1415][VER.SUI.ILS-1419][VER.SUI.NOUS-1419]
6	Prophet	in	2.13025375308825	[ADJ.0-20][NOM.A.F.S-836][NOM.A.M.S-88][NOM.A.X.S-277][NOM.D.F.S-836][NOM.D.M.S-88][NOM.D.X.S-277][NOM.G.F.S-838] [NOM.N.F.S-840][NOM.N.M.S-89][NOM.N.X.S-277][VER.IM.TU-16][VER.IN-29][VER.PI.JE-43][VER.ZU-28]
6	Prophet	innen	1.54700443675398	[NOM.A.F.P-818][NOM.D.F.P-819][NOM.G.F.P-818][NOM.N.F.P-821][VER.IN-24][VER.PI.ILS-24][VER.PI.NOUS-24] [VER.SUP.ILS-24][VER.SUP.NOUS-24]
6	Trigra	mm	1.80069685428684	[NOM.A.M.S-18][NOM.A.X.S-55][NOM.D.M.S-18][NOM.D.X.S-55][NOM.N.M.S-18][NOM.N.X.S-55][VER.II.IL-19][VER.II.JE-19] [VER.IM.TU-248][VER.PI.JE-196]
6	Trigra	mme	2.29846819083724	[NOM.A.F.S-24][NOM.A.M.P-15][NOM.A.X.P-46][NOM.D.F.S-23][NOM.G.F.S-23][NOM.G.M.P-13][NOM.G.X.P-46] [NOM.N.F.S-24][NOM.N.M.P-15][NOM.N.X.P-46][VER.IM.TU-201][VER.PI.JE-200][VER.SUI.IL-33][VER.SUI.JE-33] [VER.SUP.IL-200][VER.SUP.JE-200]
6	Trigra	mmen	2.65833259767924	[ADJ.0-11][NOM.A.F.P-26][NOM.A.X.P-14][NOM.A.X.S-17][NOM.D.F.P-27][NOM.D.M.P-18][NOM.D.X.P-60][NOM.D.X.S-17] [NOM.G.F.P-26][NOM.G.X.P-14][NOM.N.F.P-26][NOM.N.X.P-14][NOM.N.X.S-17][VER.II.ILS-19][VER.II.NOUS-19][VER.IN-200] [VER.PI.ILS-200][VER.PI.NOUS-200][VER.PPS.0-173][VER.SUI.ILS-33][VER.SUI.NOUS-33][VER.SUP.ILS-200][VER.SUP.NOUS-200] [VER.ZU-152]
6	Angloamerikan	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]

6	Angloamerikan	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
6	Angloamerikan	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
6	Angloamerikan	isches	0.0646779394943541	[ADJ.ES-1077][NOM.G.M.S-13]
6	Gouverneu	er	2.67114056179238	[ADJ-15][ADJ.0-63][ADJ.COMP.0-3865][ADJ.COMP.ER-3891][ADJ.ER-5081][ADJ.SUPER.ER-4137][NOM.A.F.P-11][NOM.A.F.S-87] [NOM.A.M.P-2646][NOM.A.M.S-2739][NOM.A.MX.P-44][NOM.A.MX.S-45][NOM.A.X.P-230][NOM.A.X.S-214][NOM.D.F.S-89] [NOM.D.M.S-2740][NOM.D.MX.S-45][NOM.D.X.P-12][NOM.D.X.S-214][NOM.G.F.P-284][NOM.G.F.S-93][NOM.G.M.P-2947] [NOM.G.MX.P-44][NOM.G.X.P-230][NOM.N.F.P-11][NOM.N.F.S-87][NOM.N.M.P-2647][NOM.N.M.S-3045][NOM.N.MX.P-44] [NOM.N.MX.S-45][NOM.N.X.P-231][NOM.N.X.S-215][VER.IM.TU-1427][VER.PI.JE-1426][VER.PPR.ER-8989][VER.PPS.ER-9074]
6	Gouverneu	rin	1.53085020685717	[NOM.A.F.S-583][NOM.A.X.S-29][NOM.D.F.S-583][NOM.D.X.S-29][NOM.G.F.S-583][NOM.N.F.S-583][NOM.N.X.S-29]
6	Bayes	sche	1.93530962289556	[ADJ.E-1092][NOM.A.F.S-88][NOM.A.M.P-63][NOM.D.F.S-83][NOM.G.F.S-83][NOM.G.M.P-58][NOM.N.F.S-88][NOM.N.M.P-63] [VER.IM.TU-155][VER.PI.JE-160][VER.SUI.IL-12][VER.SUI.JE-12][VER.SUP.IL-160][VER.SUP.JE-160]
6	Bayes	schen	2.25777783102144	[ADJ.EN-1092][NOM.A.F.P-93][NOM.A.M.P-13][NOM.A.M.S-12][NOM.A.X.S-11][NOM.D.F.P-93][NOM.D.M.P-69][NOM.D.M.S-12] [NOM.D.X.P-12][NOM.D.X.S-14][NOM.G.F.P-93][NOM.G.M.P-13][NOM.N.F.P-93][NOM.N.M.P-13][NOM.N.X.S-11][VER.II.ILS-12] [VER.II.NOUS-12][VER.IN-159][VER.PI.ILS-160][VER.PI.NOUS-160][VER.PPS.0-12][VER.SUI.ILS-12][VER.SUI.NOUS-12] [VER.SUP.ILS-160][VER.SUP.NOUS-160][VER.ZU-71]
6	Bayes	scher	1.16675840442999	[ADJ.COMP.0-910][ADJ.ER-1095][NOM.A.M.P-40][NOM.A.M.S-42][NOM.D.M.S-42][NOM.G.M.P-45][NOM.N.M.P-40][NOM.N.M.S-47]
6	Bayes	sches	0.305620250653627	[ADJ.ES-1092][NOM.G.M.S-76][NOM.G.X.S-15]
6	Mystik	al	1.90572790719735	[ADJ.0-260][NOM.A.M.S-58][NOM.A.X.S-98][NOM.D.M.S-58][NOM.D.X.S-98][NOM.N.M.S-58][NOM.N.X.S-98][VER.IM.TU-15][VER.PI.JE-15]
6	Mystik	erin	1.38629436111989	[NOM.A.F.S-534][NOM.D.F.S-534][NOM.G.F.S-534][NOM.N.F.S-534]
6	Mystik	erinnen	1.38629401750191	[NOM.A.F.P-522][NOM.D.F.P-523][NOM.G.F.P-522][NOM.N.F.P-522]
6	Mozarab	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
6	Mozarab	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
6	Mozarab	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
6	Mozarab	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
6	Aitol	ern	1.98155876486635	[ADJ.0-18][NOM.A.F.P-72][NOM.D.F.P-75][NOM.D.M.P-2657][NOM.D.M.S-11][NOM.D.MX.P-44][NOM.D.X.P-239] [NOM.G.F.P-72][NOM.N.F.P-72][VER.IM.TU-16][VER.IN-582][VER.PI.ILS-582][VER.PI.JE-16][VER.PI.NOUS-582] [VER.SUP.ILS-582][VER.SUP.NOUS-582][VER.ZU-202]
6	Aitol	ien	2.00515936981159	[ADJ.EN-19][NOM.A.F.P-1987][NOM.A.M.P-18][NOM.A.X.P-364][NOM.A.X.S-11][NOM.D.F.P-1989][NOM.D.M.P-29] [NOM.D.X.P-366][NOM.D.X.S-11][NOM.G.F.P-1987][NOM.G.F.S-44][NOM.G.M.P-17][NOM.G.X.P-364][NOM.N.F.P-1987] [NOM.N.M.P-18][NOM.N.X.P-364][NOM.N.X.S-11][VER.IN-24][VER.PI.ILS-23][VER.PI.NOUS-23][VER.SUP.ILS-53] [VER.SUP.NOUS-53][VER.ZU-12]
6	Aitol	ier	1.76973731216616	[ADJ.ER-17][NOM.A.M.P-82][NOM.A.M.S-133][NOM.A.X.S-36][NOM.D.M.S-133][NOM.D.X.S-36][NOM.G.M.P-82] [NOM.N.M.P-82][NOM.N.M.S-134][NOM.N.X.S-36][VER.IM.TU-842][VER.PI.JE-841]
6	Aitol	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
6	Aitol	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
6	Arisi	erungen	1.64575686105516	[NOM.A.F.P-510][NOM.A.M.P-40][NOM.D.F.P-510][NOM.D.M.P-41][NOM.G.F.P-510][NOM.G.M.P-39][NOM.N.F.P-511][NOM.N.M.P-39]
6	Angreif	erin	1.38629436111989	[NOM.A.F.S-534][NOM.D.F.S-534][NOM.G.F.S-534][NOM.N.F.S-534]

Prés. Rac.	Racine	Terminaison	Score d'entropie	Description
7	plebej	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
7	plebej	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
7	plebej	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
7	subordini	erende	0	[VER.PPR.E-848]
7	verkoh	lt	1.93831221127195	[ADJ.0-25][NOM.A.F.S-12][NOM.A.M.S-31][NOM.A.X.S-15][NOM.D.F.S-12][NOM.D.M.S-31][NOM.D.X.S-15][NOM.G.F.S-12] [NOM.N.F.S-12][NOM.N.M.S-31][NOM.N.X.S-15][VER.II.IL-78][VER.II.JE-78][VER.II.VOUS-64][VER.IM.TU-112][VER.IM.VOUS-1134] [VER.IM.VOUS.BVPV-19][VER.PI.IL-1187][VER.PI.JE-107][VER.PI.VOUS-1132][VER.PPS.0-1039][VER.PPS.E-25]
7	verkoh	lte	1.98555656726331	[ADJ.E-25][NOM.A.F.S-17][NOM.A.M.P-30][NOM.D.F.S-11][NOM.G.F.S-11][NOM.G.M.P-24][NOM.N.F.S-17][NOM.N.M.P-30][VER.II.IL-1075] [VER.II.JE-1073][VER.IM.TU-107][VER.PI.JE-113][VER.PPS.E-1061][VER.SUI.IL-1160][VER.SUI.JE-1158][VER.SUP.IL-113][VER.SUP.JE-113]
7	verkoh	ltem	0.109492004016052	[ADJ.EM-25][VER.PPS.EM-1062]
7	verkoh	lten	2.11744621960334	[ADJ.EN-25][NOM.A.F.P-25][NOM.D.F.P-24][NOM.D.M.P-32][NOM.D.X.P-11][NOM.G.F.P-24][NOM.N.F.P-24][VER.II.ILS-1152] [VER.II.NOUS-1152][VER.IN-113][VER.PI.ILS-113][VER.PI.NOUS-113][VER.PPS.0-83][VER.PPS.EN-1061][VER.SUI.ILS-1158] [VER.SUI.NOUS-1158][VER.SUP.ILS-113][VER.SUP.NOUS-113][VER.ZU-87]
7	verkoh	lter	0.742151061194159	[ADJ.COMP.0-19][ADJ.ER-25][NOM.A.M.P-22][NOM.A.M.S-23][NOM.D.M.S-23][NOM.G.M.P-28][NOM.N.M.P-22][NOM.N.M.S-29] [VER.PPS.ER-1063]
7	verkoh	ltes	0.290841358093647	[ADJ.ES-25][NOM.G.M.S-29][NOM.G.X.S-14][VER.PPS.ES-1063]
7	historisi	erend	0	[VER.PPR.0-848]
7	historisi	erende	0	[VER.PPR.E-848]
7	historisi	erendem	0	[VER.PPR.EM-848]
7	historisi	erenden	0	[VER.PPR.EN-848]
7	historisi	erender	0	[VER.PPR.ER-848]
7	historisi	erendes	0	[VER.PPR.ES-848]
7	bewurz	eln	2.37083238590262	[NOM.A.F.P-180][NOM.A.X.P-14][NOM.D.F.P-180][NOM.D.M.P-335][NOM.D.X.P-141][NOM.G.F.P-180][NOM.G.X.P-14][NOM.N.F.P-180] [NOM.N.X.P-14][VER.IN-608][VER.PI.ILS-606][VER.PI.NOUS-606][VER.SUP.ILS-605][VER.SUP.NOUS-605][VER.ZU-212]
7	bewurz	elt	1.70276408265721	[ADJ.0-12][VER.II.IL-73][VER.II.JE-73][VER.II.VOUS-43][VER.IM.VOUS-620][VER.IM.VOUS.BVPV-15][VER.PI.IL-634][VER.PI.VOUS-632] [VER.PPS.0-599][VER.PPS.E-25]
7	bewurz	elte	1.62596485745359	[ADJ.E-12][VER.II.IL-634][VER.II.JE-632][VER.PPS.E-622][VER.SUI.IL-707][VER.SUI.JE-705]
7	bewurz	elten	1.62579855939805	[ADJ.EN-12][VER.II.ILS-705][VER.II.NOUS-705][VER.PPS.EN-622][VER.SUI.ILS-705][VER.SUI.NOUS-705]
7	bewurz	elter	0.0938350467381319	[ADJ.ER-12][VER.PPS.ER-622]
7	bewurz	eltes	0.0938350467381319	[ADJ.ES-12][VER.PPS.ES-622]
7	Peloponnes	ier	1.76973731216616	[ADJ.ER-17][NOM.A.M.P-82][NOM.A.M.S-133][NOM.A.X.S-36][NOM.D.M.S-133][NOM.D.X.S-36][NOM.G.M.P-82][NOM.N.M.P-82] [NOM.N.M.S-134][NOM.N.X.S-36][VER.IM.TU-842][VER.PI.JE-841]
7	Peloponnes	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
7	Peloponnes	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
7	Peloponnes	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
7	Homburg	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]

7	Homburg	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
7	Geograph	ers	0.767836187084297	[NOM.A.M.P-52][NOM.A.X.P-19][NOM.D.M.P-51][NOM.D.X.P-19][NOM.G.M.P-52][NOM.G.M.S-2744][NOM.G.MX.S-45][NOM.G.X.P-19] [NOM.G.X.S-214][NOM.N.M.P-52][NOM.N.X.P-19]
7	Geograph	in	2.13025375308825	[ADJ.0-20][NOM.A.F.S-836][NOM.A.M.S-88][NOM.A.X.S-277][NOM.D.F.S-836][NOM.D.M.S-88][NOM.D.X.S-277][NOM.G.F.S-838] [NOM.N.F.S-840][NOM.N.M.S-89][NOM.N.X.S-277][VER.IM.TU-16][VER.IN-29][VER.PI.JE-43][VER.ZU-28]
7	Geograph	innen	1.54700443675398	[NOM.A.F.P-818][NOM.D.F.P-819][NOM.G.F.P-818][NOM.N.F.P-821][VER.IN-24][VER.PI.ILS-24][VER.PI.NOUS-24][VER.SUP.ILS-24] [VER.SUP.NOUS-24]
7	Geograph	us	1.91279080827913	[NOM.A.M.P-40][NOM.A.M.S-829][NOM.A.X.P-30][NOM.A.X.S-30][NOM.D.M.P-40][NOM.D.M.S-829][NOM.D.X.P-30][NOM.D.X.S-30] [NOM.G.M.P-40][NOM.G.M.S-863][NOM.G.X.P-30][NOM.G.X.S-56][NOM.N.M.P-40][NOM.N.M.S-829][NOM.N.X.P-30][NOM.N.X.S-30] [VER.IM.TU-24][VER.PI.JE-24]
7	Mandschur	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
7	Mandschur	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
7	Mandschur	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
7	Soziolog	in	2.13025375308825	[ADJ.0-20][NOM.A.F.S-836][NOM.A.M.S-88][NOM.A.X.S-277][NOM.D.F.S-836][NOM.D.M.S-88][NOM.D.X.S-277][NOM.G.F.S-838] [NOM.N.F.S-840][NOM.N.M.S-89][NOM.N.X.S-277][VER.IM.TU-16][VER.IN-29][VER.PI.JE-43][VER.ZU-28]
7	Soziolog	innen	1.54700443675398	[NOM.A.F.P-818][NOM.D.F.P-819][NOM.G.F.P-818][NOM.N.F.P-821][VER.IN-24][VER.PI.ILS-24][VER.PI.NOUS-24][VER.SUP.ILS-24] [VER.SUP.NOUS-24]
7	Lothring	ens	0.955343796057272	[NOM.A.X.S-18][NOM.D.X.S-18][NOM.G.M.S-251][NOM.G.X.S-426][NOM.N.X.S-18]
7	Lothring	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
7	Lothring	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
7	Lothring	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
7	Lothring	isches	0.0646779394943541	[ADJ.ES-1077][NOM.G.M.S-13]
7	Gegenpo	l	2.88601031767961	[ADJ.0-487][NOM.A.F.S-219][NOM.A.M.P-337][NOM.A.M.S-580][NOM.A.X.P-145][NOM.A.X.S-518][NOM.D.F.S-219][NOM.D.M.S-580] [NOM.D.X.S-518][NOM.G.F.S-220][NOM.G.M.P-336][NOM.G.X.P-145][NOM.N.F.S-221][NOM.N.M.P-337][NOM.N.M.S-585][NOM.N.X.P-145] [NOM.N.X.S-518][VER.II.IL-64][VER.II.JE-64][VER.IM.TU-823][VER.PI.IL-20][VER.PI.JE-522]
7	Gegenpo	le	2.64423867983132	[ADJ.E-484][NOM.A.F.P-26][NOM.A.F.S-327][NOM.A.M.P-178][NOM.A.M.S-13][NOM.A.X.P-191][NOM.A.X.S-70][NOM.D.F.S-313] [NOM.D.M.S-13][NOM.D.X.S-70][NOM.G.F.S-313][NOM.G.M.P-161][NOM.G.X.P-191][NOM.G.X.S-21][NOM.N.F.P-26][NOM.N.F.S-327] [NOM.N.M.P-178][NOM.N.M.S-58][NOM.N.X.P-191][NOM.N.X.S-71][VER.IM.TU-1155][VER.PI.JE-1126][VER.SUI.IL-73][VER.SUI.JE-73] [VER.SUP.IL-1148][VER.SUP.JE-1146]
7	Beweis	ung	1.63716246998186	[NOM.A.F.S-2220][NOM.A.M.S-169][NOM.D.F.S-2221][NOM.D.M.S-169][NOM.G.F.S-2220][NOM.G.M.S-152][NOM.N.F.S-2223] [NOM.N.M.S-168]
7	Sokrat	iker	1.79175946922805	[NOM.A.M.P-176][NOM.A.M.S-176][NOM.D.M.S-176][NOM.G.M.P-176][NOM.N.M.P-176][NOM.N.M.S-176]
7	Sokrat	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
7	Sokrat	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
7	Sokrat	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
Prés. Rac.	Racine	Terminaison	Score d'entropie	Description
8	neoklass	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
8	neoklass	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
8	neoklass	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]

8	neoklass	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
8	diversifizi	eren	1.82872027498886	[ADJ.COMP.EN-3886][ADJ.EN-58][NOM.A.F.P-62][NOM.A.M.P-15][NOM.A.M.S-11][NOM.D.F.P-63][NOM.D.M.P-62][NOM.D.M.S-11] [NOM.D.X.P-33][NOM.G.F.P-62][NOM.G.M.P-15][NOM.G.M.S-11][NOM.N.F.P-62][NOM.N.M.P-15][VER.IN-848][VER.PI.ILS-849] [VER.PI.NOUS-849][VER.SUP.ILS-849][VER.SUP.NOUS-849][VER.ZU-80]
8	diversifizi	erte	1.77709734163243	[ADJ.E-91][NOM.A.F.P-20][NOM.A.F.S-23][NOM.A.M.P-28][NOM.N.F.P-20][NOM.N.F.S-23][NOM.N.M.P-28][NOM.N.M.S-19] [VER.II.IL-1419][VER.II.JE-1419][VER.IM.TU-11][VER.PPS.E-1415][VER.SUI.IL-1419][VER.SUI.JE-1419]
8	diversifizi	erten	1.85574533708063	[ADJ.EN-91][NOM.A.F.P-27][NOM.A.M.P-19][NOM.A.M.S-19][NOM.D.F.P-27][NOM.D.F.S-19][NOM.D.M.P-29][NOM.D.M.S-19] [NOM.G.F.P-27][NOM.G.F.S-20][NOM.G.M.P-19][NOM.G.M.S-19][NOM.N.F.P-27][NOM.N.M.P-19][VER.II.ILS-1419][VER.II.NOUS-1419] [VER.PPS.EN-1415][VER.SUI.ILS-1419][VER.SUI.NOUS-1419]
8	diversifizi	erter	0.603150906406462	[ADJ.COMP.0-76][ADJ.ER-91][NOM.G.F.P-19][NOM.G.M.P-23][NOM.N.M.S-23][VER.PPS.ER-1415]
8	diversifizi	ertes	0.228135236147046	[ADJ.ES-91][VER.PPS.ES-1415]
8	elabori	erte	1.77709734163243	[ADJ.E-91][NOM.A.F.P-20][NOM.A.F.S-23][NOM.A.M.P-28][NOM.N.F.P-20][NOM.N.F.S-23][NOM.N.M.P-28][NOM.N.M.S-19] [VER.II.IL-1419][VER.II.JE-1419][VER.IM.TU-11][VER.PPS.E-1415][VER.SUI.IL-1419][VER.SUI.JE-1419]
8	elabori	erten	1.85574533708063	[ADJ.EN-91][NOM.A.F.P-27][NOM.A.M.P-19][NOM.A.M.S-19][NOM.D.F.P-27][NOM.D.F.S-19][NOM.D.M.P-29][NOM.D.M.S-19] [NOM.G.F.P-27][NOM.G.F.S-20][NOM.G.M.P-19][NOM.G.M.S-19][NOM.N.F.P-27][NOM.N.M.P-19][VER.II.ILS-1419][VER.II.NOUS-1419] [VER.PPS.EN-1415][VER.SUI.ILS-1419][VER.SUI.NOUS-1419]
8	elabori	erter	0.603150906406462	[ADJ.COMP.0-76][ADJ.ER-91][NOM.G.F.P-19][NOM.G.M.P-23][NOM.N.M.S-23][VER.PPS.ER-1415]
8	ortskund	ige	2.02535584020433	[ADJ.E-748][NOM.A.F.P-37][NOM.A.F.S-52][NOM.A.M.P-55][NOM.D.F.S-15][NOM.G.F.S-15][NOM.N.F.P-37][NOM.N.F.S-52][NOM.N.M.P-55] [NOM.N.M.S-47][VER.IM.TU-194][VER.PI.JE-187][VER.SUP.IL-187][VER.SUP.JE-187]
8	ortskund	igen	2.50197765093382	[ADJ.EN-748][NOM.A.F.P-62][NOM.A.M.P-48][NOM.A.M.S-48][NOM.D.F.P-62][NOM.D.F.S-37][NOM.D.M.P-56][NOM.D.M.S-48] [NOM.G.F.P-62][NOM.G.F.S-38][NOM.G.M.P-48][NOM.G.M.S-47][NOM.N.F.P-62][NOM.N.M.P-48][VER.IN-187][VER.PI.ILS-187] [VER.PI.NOUS-187][VER.SUP.ILS-187][VER.SUP.NOUS-187][VER.ZU-49]
8	ortskund	iger	1.44402118467646	[ADJ.COMP.0-563][ADJ.ER-747][NOM.A.M.P-36][NOM.A.M.S-36][NOM.D.M.S-36][NOM.G.F.P-38][NOM.G.M.P-83][NOM.N.M.P-36] [NOM.N.M.S-82]
8	sumer	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
8	sumer	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
8	sumer	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
8	sumer	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
8	seelsorger	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
8	seelsorger	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
8	seelsorger	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
8	seelsorger	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
8	seelsorger	lich	0.620187444621208	[ADJ.0-468][VER.II.IL-27][VER.II.JE-27][VER.IM.TU-14][VER.PI.JE-14]
8	seelsorger	liche	0.817626526609839	[ADJ.E-468][VER.IM.TU-14][VER.PI.JE-14][VER.SUI.IL-27][VER.SUI.JE-27][VER.SUP.IL-14][VER.SUP.JE-14]
8	magyar	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
8	magyar	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
8	magyar	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
8	magyar	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
8	mitbew	egen	2.33276006310028	[NOM.A.F.P-13][NOM.D.F.P-13][NOM.D.M.P-27][NOM.G.F.P-13][NOM.N.F.P-13][VER.II.ILS-31][VER.II.NOUS-31][VER.IN-210][VER.PI.ILS-210][VER.PI.NOUS-210][VER.PPS.0-76][VER.SUI.ILS-31][VER.SUI.NOUS-31][VER.SUP.ILS-210][VER.SUP.NOUS-210][VER.ZU-169]
8	mitbew	egt	1.46884951491177	[VER.II.VOUS-31][VER.IM.VOUS-214][VER.PI.IL-210][VER.PI.VOUS-210][VER.PPS.0-110]
8	mitbew	egte	1.6094312646213	[VER.II.IL-110][VER.II.JE-110][VER.PPS.E-109][VER.SUI.IL-110][VER.SUI.JE-110]
8	mitbew	egten	1.6094312646213	[VER.II.ILS-110][VER.II.NOUS-110][VER.PPS.EN-109][VER.SUI.ILS-110][VER.SUI.NOUS-110]
8	logarithmi	erte	1.77709734163243	[ADJ.E-91][NOM.A.F.P-20][NOM.A.F.S-23][NOM.A.M.P-28][NOM.N.F.P-20][NOM.N.F.S-23][NOM.N.M.P-28][NOM.N.M.S-19] [VER.II.IL-1419][VER.II.JE-1419][VER.IM.TU-11][VER.PPS.E-1415][VER.SUI.IL-1419][VER.SUI.JE-1419]
8	logarithmi	erten	1.85574533708063	[ADJ.EN-91][NOM.A.F.P-27][NOM.A.M.P-19][NOM.A.M.S-19][NOM.D.F.P-27][NOM.D.F.S-19][NOM.D.M.P-29][NOM.D.M.S-19] [NOM.G.F.P-27][NOM.G.F.S-20][NOM.G.M.P-19][NOM.G.M.S-19][NOM.N.F.P-27][NOM.N.M.P-19][VER.II.ILS-1419][VER.II.NOUS-1419]

				[VER.PPS.EN-1415][VER.SUI.ILS-1419][VER.SUI.NOUS-1419]
8	faktori	ellen	2.4832396631456	[ADJ.EN-81][NOM.A.F.P-68][NOM.D.F.P-68][NOM.D.M.P-16][NOM.D.X.P-21][NOM.G.F.P-68][NOM.N.F.P-68][VER.IN-97][VER.PI.ILS-97] [VER.PI.NOUS-97][VER.SUP.ILS-97][VER.SUP.NOUS-97][VER.ZU-74]
8	faktori	sieren	1.6094379124341	[VER.IN-160][VER.PI.ILS-160][VER.PI.NOUS-160][VER.SUP.ILS-160][VER.SUP.NOUS-160]
8	faktori	siert	1.38629436111989	[VER.IM.VOUS-160][VER.PI.IL-160][VER.PI.VOUS-160][VER.PPS.0-160]
8	Kirgis	ien	2.00515936981159	[ADJ.EN-19][NOM.A.F.P-1987][NOM.A.M.P-18][NOM.A.X.P-364][NOM.A.X.S-11][NOM.D.F.P-1989][NOM.D.M.P-29][NOM.D.X.P-366] [NOM.D.X.S-11][NOM.G.F.P-1987][NOM.G.F.S-44][NOM.G.M.P-17][NOM.G.X.P-364][NOM.N.F.P-1987][NOM.N.M.P-18][NOM.N.X.P-364] [NOM.N.X.S-11][VER.IN-24][VER.PI.ILS-23][VER.PI.NOUS-23][VER.SUP.ILS-53][VER.SUP.NOUS-53][VER.ZU-12]
8	Kirgis	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
8	Kirgis	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
8	Kirgis	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
8	Kirgis	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
8	Laot	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
8	Laot	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
8	Laot	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
8	Laot	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
8	Laot	se	2.83970327919849	[ADJ.E-223][NOM.A.F.P-63][NOM.A.F.S-551][NOM.A.M.P-300][NOM.A.X.P-92][NOM.A.X.S-19][NOM.D.F.P-15][NOM.D.F.S-540] [NOM.D.X.S-20][NOM.G.F.P-53][NOM.G.F.S-543][NOM.G.M.P-288][NOM.G.X.P-92][NOM.N.F.P-63][NOM.N.F.S-553][NOM.N.M.P-300] [NOM.N.M.S-68][NOM.N.X.P-92][NOM.N.X.S-19][VER.IM.TU-343][VER.PI.JE-368][VER.SUI.IL-79][VER.SUI.JE-79][VER.SUP.IL-377] [VER.SUP.JE-377]
8	Klingon	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
8	Klingon	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
8	Philanthrop	in	2.13025375308825	[ADJ.0-20][NOM.A.F.S-836][NOM.A.M.S-88][NOM.A.X.S-277][NOM.D.F.S-836][NOM.D.M.S-88][NOM.D.X.S-277][NOM.G.F.S-838] [NOM.N.F.S-840][NOM.N.M.S-89][NOM.N.X.S-277][VER.IM.TU-16][VER.IN-29][VER.PI.JE-43][VER.ZU-28]
8	Philanthrop	ismus	1.38629401396632	[NOM.A.M.S-520][NOM.D.M.S-520][NOM.G.M.S-519][NOM.N.M.S-520]
8	Philanthrop	ist	1.70133750601046	[NOM.A.M.S-15][NOM.D.M.S-15][NOM.N.M.S-413][VER.IM.TU-15][VER.IM.VOUS-51][VER.PI.IL-77][VER.PI.JE-15][VER.PI.TU-97] [VER.PI.VOUS-49][VER.PPS.0-28][VER.SUP.TU-29]

Prés. Rac.	Racine	Terminaison	Score d'entropie	Description
9	arabi	siert	1.38629436111989	[VER.IM.VOUS-160][VER.PI.IL-160][VER.PI.VOUS-160][VER.PPS.0-160]
9	arabi	sierte	1.6094379124341	[VER.II.IL-160][VER.II.JE-160][VER.PPS.E-160][VER.SUI.IL-160][VER.SUI.JE-160]
9	arabi	sierten	1.6094379124341	[VER.II.ILS-160][VER.II.NOUS-160][VER.PPS.EN-160][VER.SUI.ILS-160][VER.SUI.NOUS-160]
9	zelebri	eren	1.82872027498886	[ADJ.COMP.EN-3886][ADJ.EN-58][NOM.A.F.P-62][NOM.A.M.P-15][NOM.A.M.S-11][NOM.D.F.P-63][NOM.D.M.P-62][NOM.D.M.S-11] [NOM.D.X.P-33][NOM.G.F.P-62][NOM.G.M.P-15][NOM.G.M.S-11][NOM.N.F.P-62][NOM.N.M.P-15][VER.IN-848][VER.PI.ILS-849] [VER.PI.NOUS-849][VER.SUP.ILS-849][VER.SUP.NOUS-849][VER.ZU-80]
9	zelebri	erenden	0	[VER.PPR.EN-848]
9	zelebri	erte	1.77709734163243	[ADJ.E-91][NOM.A.F.P-20][NOM.A.F.S-23][NOM.A.M.P-28][NOM.N.F.P-20][NOM.N.F.S-23][NOM.N.M.P-28][NOM.N.M.S-19][VER.II.IL-1419] [VER.II.JE-1419][VER.IM.TU-11][VER.PPS.E-1415][VER.SUI.IL-1419][VER.SUI.JE-1419]
9	zelebri	erten	1.85574533708063	[ADJ.EN-91][NOM.A.F.P-27][NOM.A.M.P-19][NOM.A.M.S-19][NOM.D.F.P-27][NOM.D.F.S-19][NOM.D.M.P-29][NOM.D.M.S-19] [NOM.G.F.P-27][NOM.G.F.S-20][NOM.G.M.P-19][NOM.G.M.S-19][NOM.N.F.P-27][NOM.N.M.P-19][VER.II.ILS-1419][VER.II.NOUS-1419] [VER.PPS.EN-1415][VER.SUI.ILS-1419][VER.SUI.NOUS-1419]
9	polynes	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
9	polynes	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
9	polynes	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
9	polynes	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]

9	partitioni	eren	1.82872027498886	[ADJ.COMP.EN-3886][ADJ.EN-58][NOM.A.F.P-62][NOM.A.M.P-15][NOM.A.M.S-11][NOM.D.F.P-63][NOM.D.M.P-62][NOM.D.M.S-11][NOM.D.X.P-33][NOM.G.F.P-62][NOM.G.M.P-15][NOM.G.M.S-11][NOM.N.F.P-62][NOM.N.M.P-15][VER.IN-848][VER.PI.ILS-849][VER.PI.NOUS-849][VER.SUP.ILS-849][VER.SUP.NOUS-849][VER.ZU-80]
9	partitioni	erten	1.85574533708063	[ADJ.EN-91][NOM.A.F.P-27][NOM.A.M.P-19][NOM.A.M.S-19][NOM.D.F.P-27][NOM.D.F.S-19][NOM.D.M.P-29][NOM.D.M.S-19][NOM.G.F.P-27][NOM.G.F.S-20][NOM.G.M.P-19][NOM.G.M.S-19][NOM.N.F.P-27][NOM.N.M.P-19][VER.II.ILS-1419][VER.II.NOUS-1419][VER.PPS.EN-1415][VER.SUI.ILS-1419][VER.SUI.NOUS-1419]
9	dissozi	al	1.90572790719735	[ADJ.0-260][NOM.A.M.S-58][NOM.A.X.S-98][NOM.D.M.S-58][NOM.D.X.S-98][NOM.N.M.S-58][NOM.N.X.S-98][VER.IM.TU-15][VER.PI.JE-15]
9	dissozi	alen	2.43225887195374	[ADJ.EN-259][NOM.A.F.P-40][NOM.A.M.P-24][NOM.A.M.S-23][NOM.D.F.P-43][NOM.D.M.P-68][NOM.D.M.S-23][NOM.D.X.P-69][NOM.G.F.P-40][NOM.G.M.P-24][NOM.G.M.S-23][NOM.N.F.P-40][NOM.N.M.P-24][VER.IN-15][VER.PI.ILS-15][VER.PI.NOUS-15][VER.SUP.ILS-15][VER.SUP.NOUS-15]
9	dissozi	iert	1.44570647062607	[ADJ.0-54][VER.IM.VOUS-841][VER.PI.IL-842][VER.PI.VOUS-841][VER.PPS.0-830]
9	hawai	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
9	hawai	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
9	hawai	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
9	hawai	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
9	Tyrrhen	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
9	Tyrrhen	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
9	Tyrrhen	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
9	Tyrrhen	isches	0.0646779394943541	[ADJ.ES-1077][NOM.G.M.S-13]
9	Biograph	in	2.13025375308825	[ADJ.0-20][NOM.A.F.S-836][NOM.A.M.S-88][NOM.A.X.S-277][NOM.D.F.S-836][NOM.D.M.S-88][NOM.D.X.S-277][NOM.G.F.S-838][NOM.N.F.S-840][NOM.N.M.S-89][NOM.N.X.S-277][VER.IM.TU-16][VER.IN-29][VER.PI.JE-43][VER.ZU-28]
9	Patriarch	al	1.90572790719735	[ADJ.0-260][NOM.A.M.S-58][NOM.A.X.S-98][NOM.D.M.S-58][NOM.D.X.S-98][NOM.N.M.S-58][NOM.N.X.S-98][VER.IM.TU-15][VER.PI.JE-15]
9	Patriarch	in	2.13025375308825	[ADJ.0-20][NOM.A.F.S-836][NOM.A.M.S-88][NOM.A.X.S-277][NOM.D.F.S-836][NOM.D.M.S-88][NOM.D.X.S-277][NOM.G.F.S-838][NOM.N.F.S-840][NOM.N.M.S-89][NOM.N.X.S-277][VER.IM.TU-16][VER.IN-29][VER.PI.JE-43][VER.ZU-28]
9	Erhab	ene	0.966137771583489	[ADJ.E-189][NOM.A.F.P-31][NOM.A.F.S-50][NOM.A.M.P-40][NOM.A.X.P-28][NOM.D.F.S-21][NOM.G.F.S-22][NOM.G.X.P-28][NOM.N.F.P-31][NOM.N.F.S-50][NOM.N.M.P-40][NOM.N.M.S-38][NOM.N.X.P-28][VER.PPS.E-2358]
9	Erhab	enen	1.1643626324349	[ADJ.EN-189][NOM.A.F.P-51][NOM.A.M.P-38][NOM.A.M.S-38][NOM.D.F.P-51][NOM.D.F.S-29][NOM.D.M.P-45][NOM.D.M.S-38][NOM.D.X.P-29][NOM.G.F.P-51][NOM.G.F.S-29][NOM.G.M.P-38][NOM.G.M.S-38][NOM.N.F.P-51][NOM.N.M.P-38][VER.PPS.EN-2359]
9	Erhab	ener	0.62531967045454	[ADJ.COMP.0-121][ADJ.ER-189][NOM.G.F.P-29][NOM.G.M.P-39][NOM.N.M.S-39][VER.PPS.ER-2359]
9	Erhab	enes	0.342945979351159	[ADJ.ES-189][NOM.G.X.S-42][VER.PPS.ES-2359]
9	Wallon	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
9	Wallon	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
9	Wallon	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
9	Wallon	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]

Prés. Rac.	Racine	Terminaison	Score d'entropie	Description
10	ungebor	ene	0.966137771583489	[ADJ.E-189][NOM.A.F.P-31][NOM.A.F.S-50][NOM.A.M.P-40][NOM.A.X.P-28][NOM.D.F.S-21][NOM.G.F.S-22][NOM.G.X.P-28][NOM.N.F.P-31][NOM.N.F.S-50][NOM.N.M.P-40][NOM.N.M.S-38][NOM.N.X.P-28][VER.PPS.E-2358]
10	ungebor	enen	1.1643626324349	[ADJ.EN-189][NOM.A.F.P-51][NOM.A.M.P-38][NOM.A.M.S-38][NOM.D.F.P-51][NOM.D.F.S-29][NOM.D.M.P-45][NOM.D.M.S-38][NOM.D.X.P-29][NOM.G.F.P-51][NOM.G.F.S-29][NOM.G.M.P-38][NOM.G.M.S-38][NOM.N.F.P-51][NOM.N.M.P-38][VER.PPS.EN-2359]
10	ungebor	ener	0.62531967045454	[ADJ.COMP.0-121][ADJ.ER-189][NOM.G.F.P-29][NOM.G.M.P-39][NOM.N.M.S-39][VER.PPS.ER-2359]
10	ungebor	enes	0.342945979351159	[ADJ.ES-189][NOM.G.X.S-42][VER.PPS.ES-2359]
10	algebra	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
10	algebra	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]

10	algebra	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
10	algebra	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
10	algebra	isches	0.0646779394943541	[ADJ.ES-1077][NOM.G.M.S-13]
10	algebra	s	2.69537712595293	[ADJ.0-216][ADJ.COMP.ES-3892][ADJ.ES-5072][ADJ.SUPER.ES-4136][NOM.A.F.P-329][NOM.A.F.S-305][NOM.A.M.P-1055] [NOM.A.M.S-1087][NOM.A.X.P-873][NOM.A.X.S-235][NOM.D.F.P-329][NOM.D.F.S-305][NOM.D.M.P-1049][NOM.D.M.S-1087][NOM.D.X.P-873][NOM.D.X.S-191][NOM.G.F.P-329][NOM.G.F.S-330][NOM.G.M.P-1055][NOM.G.M.S-11174][NOM.G.MX.S-60][NOM.G.X.P-873] [NOM.G.X.S-7538][NOM.N.F.P-330][NOM.N.F.S-306][NOM.N.M.P-1052][NOM.N.M.S-1089][NOM.N.X.P-873][NOM.N.X.S-235][VER.II.IL-72] [VER.II.JE-72][VER.IM.TU-214][VER.PI.JE-211][VER.PPR.ES-8989][VER.PPS.ES-9076]
10	kollabi	eren	1.82872027498886	[ADJ.COMP.EN-3886][ADJ.EN-58][NOM.A.F.P-62][NOM.A.M.P-15][NOM.A.M.S-11][NOM.D.F.P-63][NOM.D.M.P-62][NOM.D.M.S-11] [NOM.D.X.P-33][NOM.G.F.P-62][NOM.G.M.P-15][NOM.G.M.S-11][NOM.N.F.P-62][NOM.N.M.P-15][VER.IN-848][VER.PI.ILS-849] [VER.PI.NOUS-849][VER.SUP.ILS-849][VER.SUP.NOUS-849][VER.ZU-80]
10	kollabi	erenden	0	[VER.PPR.EN-848]
10	kollabi	erte	1.77709734163243	[ADJ.E-91][NOM.A.F.P-20][NOM.A.F.S-23][NOM.A.M.P-28][NOM.N.F.P-20][NOM.N.F.S-23][NOM.N.M.P-28][NOM.N.M.S-19][VER.II.IL-1419] [VER.II.JE-1419][VER.IM.TU-11][VER.PPS.E-1415][VER.SUI.IL-1419][VER.SUI.JE-1419]
10	kollabi	erten	1.85574533708063	[ADJ.EN-91][NOM.A.F.P-27][NOM.A.M.P-19][NOM.A.M.S-19][NOM.D.F.P-27][NOM.D.F.S-19][NOM.D.M.P-29][NOM.D.M.S-19] [NOM.G.F.P-27][NOM.G.F.S-20][NOM.G.M.P-19][NOM.G.M.S-19][NOM.N.F.P-27][NOM.N.M.P-19][VER.II.ILS-1419][VER.II.NOUS-1419] [VER.PPS.EN-1415][VER.SUI.ILS-1419][VER.SUI.NOUS-1419]
10	bedeutsam	ere	1.76448550779455	[ADJ.COMP.E-3884][ADJ.E-58][NOM.A.F.S-49][NOM.A.M.P-52][NOM.A.X.P-31][NOM.A.X.S-15][NOM.D.F.S-46][NOM.G.F.S-47] [NOM.G.M.P-47][NOM.G.X.P-31][NOM.N.F.S-49][NOM.N.M.P-52][NOM.N.M.S-16][NOM.N.X.P-31][NOM.N.X.S-15][VER.IM.TU-1435] [VER.PI.JE-1431][VER.SUP.IL-1431][VER.SUP.JE-1431]
10	bedeutsam	es	2.05613203231168	[ADJ.COMP.ES-3892][ADJ.ES-5072][ADJ.SUPER.ES-4136][NOM.A.F.P-91][NOM.A.F.S-11][NOM.A.M.P-111][NOM.A.M.S-18][NOM.A.X.P-107] [NOM.A.X.S-55][NOM.D.F.P-91][NOM.D.F.S-11][NOM.D.M.P-110][NOM.D.M.S-18][NOM.D.X.P-107][NOM.D.X.S-11][NOM.G.F.P-91] [NOM.G.F.S-22][NOM.G.M.P-111][NOM.G.M.S-3029][NOM.G.X.P-107][NOM.G.X.S-2477][NOM.N.F.P-91][NOM.N.F.S-11][NOM.N.M.P-111] [NOM.N.M.S-18][NOM.N.X.P-107][NOM.N.X.S-55][VER.II.IL-31][VER.II.JE-31][VER.IM.TU-25][VER.PI.JE-25][VER.PPR.ES-8989] [VER.PPS.ES-9076]
10	bedeutsam	ste	1.4071452473111	[ADJ.E-28][ADJ.SUPER.E-4136][NOM.A.F.P-20][NOM.A.F.S-49][NOM.A.M.P-75][NOM.A.X.P-16][NOM.D.F.S-42][NOM.G.F.P-13][NOM.G.F.S-43][NOM.G.M.P-65][NOM.G.X.P-15][NOM.N.F.P-20][NOM.N.F.S-49][NOM.N.M.P-75][NOM.N.M.S-12][NOM.N.X.P-16][VER.II.IL-143] [VER.II.JE-143][VER.IM.TU-88][VER.PI.JE-85][VER.PPS.E-143][VER.SUI.IL-144][VER.SUI.JE-144][VER.SUP.IL-85][VER.SUP.JE-85]
10	bedeutsam	sten	2.18646870363207	[ADJ.EN-28][ADJ.SUPER.EN-4135][NOM.A.F.P-70][NOM.A.M.P-437][NOM.A.M.S-437][NOM.D.F.P-81][NOM.D.M.P-501][NOM.D.M.S-437] [NOM.D.X.P-17][NOM.G.F.P-70][NOM.G.M.P-437][NOM.G.M.S-431][NOM.N.F.P-70][NOM.N.M.P-437][VER.II.ILS-144][VER.II.NOUS-144] [VER.IN-85][VER.PI.ILS-85][VER.PI.NOUS-85][VER.PPS.EN-143][VER.SUI.ILS-144][VER.SUI.NOUS-144][VER.SUP.ILS-85] [VER.SUP.NOUS-85][VER.ZU-34]
10	bedeutsam	stes	0.298342442587722	[ADJ.ES-27][ADJ.SUPER.ES-4135][NOM.G.M.S-75][NOM.G.X.S-23][VER.PPS.ES-143]
10	aggregi	eren	1.82872027498886	[ADJ.COMP.EN-3886][ADJ.EN-58][NOM.A.F.P-62][NOM.A.M.P-15][NOM.A.M.S-11][NOM.D.F.P-63][NOM.D.M.P-62][NOM.D.M.S-11] [NOM.D.X.P-33][NOM.G.F.P-62][NOM.G.M.P-15][NOM.G.M.S-11][NOM.N.F.P-62][NOM.N.M.P-15][VER.IN-848][VER.PI.ILS-849] [VER.PI.NOUS-849][VER.SUP.ILS-849][VER.SUP.NOUS-849][VER.ZU-80]
10	aggregi	erte	1.77709734163243	[ADJ.E-91][NOM.A.F.P-20][NOM.A.F.S-23][NOM.A.M.P-28][NOM.N.F.P-20][NOM.N.F.S-23][NOM.N.M.P-28][NOM.N.M.S-19][VER.II.IL-1419] [VER.II.JE-1419][VER.IM.TU-11][VER.PPS.E-1415][VER.SUI.IL-1419][VER.SUI.JE-1419]
10	aggregi	erten	1.85574533708063	[ADJ.EN-91][NOM.A.F.P-27][NOM.A.M.P-19][NOM.A.M.S-19][NOM.D.F.P-27][NOM.D.F.S-19][NOM.D.M.P-29][NOM.D.M.S-19] [NOM.G.F.P-27][NOM.G.F.S-20][NOM.G.M.P-19][NOM.G.M.S-19][NOM.N.F.P-27][NOM.N.M.P-19][VER.II.ILS-1419][VER.II.NOUS-1419] [VER.PPS.EN-1415][VER.SUI.ILS-1419][VER.SUI.NOUS-1419]
10	aggregi	erter	0.603150906406462	[ADJ.COMP.0-76][ADJ.ER-91][NOM.G.F.P-19][NOM.G.M.P-23][NOM.N.M.S-23][VER.PPS.ER-1415]
10	brack	ig	1.00053136036751	[ADJ.0-746][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-171][VER.PI.JE-172]
10	brack	ige	2.02535584020433	[ADJ.E-748][NOM.A.F.P-37][NOM.A.F.S-52][NOM.A.M.P-55][NOM.D.F.S-15][NOM.G.F.S-15][NOM.N.F.P-37][NOM.N.F.S-52][NOM.N.M.P-55] [NOM.N.M.S-47][VER.IM.TU-194][VER.PI.JE-187][VER.SUP.IL-187][VER.SUP.JE-187]
10	brack	igen	2.50197765093382	[ADJ.EN-748][NOM.A.F.P-62][NOM.A.M.P-48][NOM.A.M.S-48][NOM.D.F.P-62][NOM.D.F.S-37][NOM.D.M.P-56][NOM.D.M.S-48] [NOM.G.F.P-62][NOM.G.F.S-38][NOM.G.M.P-48][NOM.G.M.S-47][NOM.N.F.P-62][NOM.N.M.P-48][VER.IN-187][VER.PI.ILS-187]

				[VER.PI.NOUS-187][VER.SUP.ILS-187][VER.SUP.NOUS-187][VER.ZU-49]
10	brack	iges	0	[ADJ.ES-746]
10	Kaliforni	a	2.62774290576343	[NOM.A.F.P-29][NOM.A.F.S-394][NOM.A.M.S-71][NOM.A.X.P-353][NOM.A.X.S-150][NOM.D.F.P-29][NOM.D.F.S-394][NOM.D.M.S-71] [NOM.D.X.P-353][NOM.D.X.S-150][NOM.G.F.P-29][NOM.G.F.S-414][NOM.G.M.S-11][NOM.G.X.P-353][NOM.G.X.S-31][NOM.N.F.P-29] [NOM.N.F.S-394][NOM.N.M.S-71][NOM.N.X.P-353][NOM.N.X.S-149]
10	Kaliforni	ens	0.955343796057272	[NOM.A.X.S-18][NOM.D.X.S-18][NOM.G.M.S-251][NOM.G.X.S-426][NOM.N.X.S-18]
10	Kaliforni	ern	1.98155876486635	[ADJ.0-18][NOM.A.F.P-72][NOM.D.F.P-75][NOM.D.M.P-2657][NOM.D.M.S-11][NOM.D.MX.P-44][NOM.D.X.P-239][NOM.G.F.P-72] [NOM.N.F.P-72][VER.IM.TU-16][VER.IN-582][VER.PI.ILS-582][VER.PI.JE-16][VER.PI.NOUS-582][VER.SUP.ILS-582][VER.SUP.NOUS-582] [VER.ZU-202]
10	Hannover	a	2.62774290576343	[NOM.A.F.P-29][NOM.A.F.S-394][NOM.A.M.S-71][NOM.A.X.P-353][NOM.A.X.S-150][NOM.D.F.P-29][NOM.D.F.S-394][NOM.D.M.S-71] [NOM.D.X.P-353][NOM.D.X.S-150][NOM.G.F.P-29][NOM.G.F.S-414][NOM.G.M.S-11][NOM.G.X.P-353][NOM.G.X.S-31][NOM.N.F.P-29] [NOM.N.F.S-394][NOM.N.M.S-71][NOM.N.X.P-353][NOM.N.X.S-149]
10	Hannover	sch	1.39614361053955	[ADJ.0-1092][NOM.A.M.S-86][NOM.A.X.S-20][NOM.D.M.S-86][NOM.D.X.S-19][NOM.N.M.S-88][NOM.N.X.S-20][VER.II.IL-12][VER.II.JE-12] [VER.IM.TU-161][VER.PI.JE-158]
10	Hannover	sche	1.93530962289556	[ADJ.E-1092][NOM.A.F.S-88][NOM.A.M.P-63][NOM.D.F.S-83][NOM.G.F.S-83][NOM.G.M.P-58][NOM.N.F.S-88][NOM.N.M.P-63] [VER.IM.TU-155][VER.PI.JE-160][VER.SUI.IL-12][VER.SUI.JE-12][VER.SUP.IL-160][VER.SUP.JE-160]
10	Hannover	schen	2.25777783102144	[ADJ.EN-1092][NOM.A.F.P-93][NOM.A.M.P-13][NOM.A.M.S-12][NOM.A.X.S-11][NOM.D.F.P-93][NOM.D.M.P-69][NOM.D.M.S-12] [NOM.D.X.P-12][NOM.D.X.S-14][NOM.G.F.P-93][NOM.G.M.P-13][NOM.N.F.P-93][NOM.N.M.P-13][NOM.N.X.S-11][VER.II.ILS-12] [VER.II.NOUS-12][VER.IN-159][VER.PI.ILS-160][VER.PI.NOUS-160][VER.PPS.0-12][VER.SUI.ILS-12][VER.SUI.NOUS-12][VER.SUP.ILS-160] [VER.SUP.NOUS-160][VER.ZU-71]
10	Hannover	scher	1.16675840442999	[ADJ.COMP.0-910][ADJ.ER-1095][NOM.A.M.P-40][NOM.A.M.S-42][NOM.D.M.S-42][NOM.G.M.P-45][NOM.N.M.P-40][NOM.N.M.S-47]
10	Hannover	sches	0.305620250653627	[ADJ.ES-1092][NOM.G.M.S-76][NOM.G.X.S-15]
Prés. Rac.	Racine	Terminaison	Score d'entropie	Description
11	malay	i	2.60182960519884	[ADJ.0-29][NOM.A.F.S-497][NOM.A.M.P-117][NOM.A.M.S-89][NOM.A.X.P-40][NOM.A.X.S-65][NOM.D.F.S-497][NOM.D.M.P-116] [NOM.D.M.S-89][NOM.D.X.P-40][NOM.D.X.S-65][NOM.G.F.S-502][NOM.G.M.P-117][NOM.G.X.P-40][NOM.N.F.S-498][NOM.N.M.P-117] [NOM.N.M.S-89][NOM.N.X.P-40][NOM.N.X.S-65][VER.IM.TU-50][VER.PI.JE-20][VER.SUP.IL-29][VER.SUP.JE-29]
11	malay	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
11	malay	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
11	malay	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
11	intendi	erte	1.77709734163243	[ADJ.E-91][NOM.A.F.P-20][NOM.A.F.S-23][NOM.A.M.P-28][NOM.N.F.P-20][NOM.N.F.S-23][NOM.N.M.P-28][NOM.N.M.S-19][VER.II.IL-1419] [VER.II.JE-1419][VER.IM.TU-11][VER.PPS.E-1415][VER.SUI.IL-1419][VER.SUI.JE-1419]
11	intendi	erten	1.85574533708063	[ADJ.EN-91][NOM.A.F.P-27][NOM.A.M.P-19][NOM.A.M.S-19][NOM.D.F.P-27][NOM.D.F.S-19][NOM.D.M.P-29][NOM.D.M.S-19] [NOM.G.F.P-27][NOM.G.F.S-20][NOM.G.M.P-19][NOM.G.M.S-19][NOM.N.F.P-27][NOM.N.M.P-19][VER.II.ILS-1419][VER.II.NOUS-1419] [VER.PPS.EN-1415][VER.SUI.ILS-1419][VER.SUI.NOUS-1419]
11	referenzi	elle	2.45275689185127	[ADJ.E-81][NOM.A.F.S-64][NOM.A.M.P-14][NOM.A.X.P-19][NOM.D.F.S-59][NOM.G.F.S-59][NOM.G.X.P-19][NOM.N.F.S-64][NOM.N.M.P-14] [NOM.N.X.P-19][VER.IM.TU-95][VER.PI.JE-97][VER.SUP.IL-97][VER.SUP.JE-97]
11	referenzi	eren	1.82872027498886	[ADJ.COMP.EN-3886][ADJ.EN-58][NOM.A.F.P-62][NOM.A.M.P-15][NOM.A.M.S-11][NOM.D.F.P-63][NOM.D.M.P-62][NOM.D.M.S-11] [NOM.D.X.P-33][NOM.G.F.P-62][NOM.G.M.P-15][NOM.G.M.S-11][NOM.N.F.P-62][NOM.N.M.P-15][VER.IN-848][VER.PI.ILS-849] [VER.PI.NOUS-849][VER.SUP.ILS-849][VER.SUP.NOUS-849][VER.ZU-80]
11	referenzi	erte	1.77709734163243	[ADJ.E-91][NOM.A.F.P-20][NOM.A.F.S-23][NOM.A.M.P-28][NOM.N.F.P-20][NOM.N.F.S-23][NOM.N.M.P-28][NOM.N.M.S-19][VER.II.IL-1419] [VER.II.JE-1419][VER.IM.TU-11][VER.PPS.E-1415][VER.SUI.IL-1419][VER.SUI.JE-1419]
11	referenzi	erten	1.85574533708063	[ADJ.EN-91][NOM.A.F.P-27][NOM.A.M.P-19][NOM.A.M.S-19][NOM.D.F.P-27][NOM.D.F.S-19][NOM.D.M.P-29][NOM.D.M.S-19] [NOM.G.F.P-27][NOM.G.F.S-20][NOM.G.M.P-19][NOM.G.M.S-19][NOM.N.F.P-27][NOM.N.M.P-19][VER.II.ILS-1419][VER.II.NOUS-1419] [VER.PPS.EN-1415][VER.SUI.ILS-1419][VER.SUI.NOUS-1419]
11	lid	u	2.29015466110329	[ADJ.0-14][NOM.A.F.S-17][NOM.A.M.S-48][NOM.A.X.S-38][NOM.D.F.S-17][NOM.D.M.S-48][NOM.D.X.S-38][NOM.G.F.S-17][NOM.N.F.S-17]

				[NOM.N.M.S-49][NOM.N.X.S-38][VER.IM.TU-129][VER.PI.JE-127]
11	kristall	enen	1.1643626324349	[ADJ.EN-189][NOM.A.F.P-51][NOM.A.M.P-38][NOM.A.M.S-38][NOM.D.F.P-51][NOM.D.F.S-29][NOM.D.M.P-45][NOM.D.M.S-38] [NOM.D.X.P-29][NOM.G.F.P-51][NOM.G.F.S-29][NOM.G.M.P-38][NOM.G.M.S-38][NOM.N.F.P-51][NOM.N.M.P-38][VER.PPS.EN-2359]
11	kristall	in	2.13025375308825	[ADJ.0-20][NOM.A.F.S-836][NOM.A.M.S-88][NOM.A.X.S-277][NOM.D.F.S-836][NOM.D.M.S-88][NOM.D.X.S-277][NOM.G.F.S-838] [NOM.N.F.S-840][NOM.N.M.S-89][NOM.N.X.S-277][VER.IM.TU-16][VER.IN-29][VER.PI.JE-43][VER.ZU-28]
11	kristall	ine	2.53238594558958	[ADJ.E-19][NOM.A.F.S-115][NOM.A.M.P-56][NOM.A.X.P-84][NOM.D.F.S-114][NOM.G.F.S-116][NOM.G.M.P-54][NOM.G.X.P-84] [NOM.N.F.S-115][NOM.N.M.P-56][NOM.N.M.S-11][NOM.N.X.P-84][VER.IM.TU-16][VER.PI.JE-16][VER.SUP.IL-16][VER.SUP.JE-16]
11	kristall	inen	2.28855985808302	[ADJ.EN-19][NOM.A.F.P-128][NOM.A.M.P-12][NOM.D.F.P-129][NOM.D.M.P-66][NOM.D.X.P-85][NOM.G.F.P-128][NOM.G.M.P-12] [NOM.N.F.P-128][NOM.N.M.P-12][VER.IN-16][VER.PI.ILS-16][VER.PI.NOUS-16][VER.SUP.ILS-16][VER.SUP.NOUS-16]
11	Fibr	a	2.62774290576343	[NOM.A.F.P-29][NOM.A.F.S-394][NOM.A.M.S-71][NOM.A.X.P-353][NOM.A.X.S-150][NOM.D.F.P-29][NOM.D.F.S-394][NOM.D.M.S-71] [NOM.D.X.P-353][NOM.D.X.S-150][NOM.G.F.P-29][NOM.G.F.S-414][NOM.G.M.S-11][NOM.G.X.P-353][NOM.G.X.S-31][NOM.N.F.P-29] [NOM.N.F.S-394][NOM.N.M.S-71][NOM.N.X.P-353][NOM.N.X.S-149]
11	Fibr	ate	2.21757685815395	[NOM.A.F.S-24][NOM.A.M.P-40][NOM.A.X.P-200][NOM.D.F.S-24][NOM.G.F.S-24][NOM.G.M.P-40][NOM.G.X.P-200][NOM.N.F.S-24] [NOM.N.M.P-40][NOM.N.X.P-200][VER.IM.TU-36][VER.PI.JE-30][VER.SUP.IL-30][VER.SUP.JE-30]
11	Fibr	o	2.11131543272431	[NOM.A.F.S-34][NOM.A.M.S-154][NOM.A.X.S-173][NOM.D.F.S-34][NOM.D.M.S-154][NOM.D.X.S-173][NOM.G.F.S-34][NOM.N.F.S-34] [NOM.N.M.S-154][NOM.N.X.S-173]
11	Fibr	ose	1.88687613472433	[ADJ.E-139][NOM.A.F.P-14][NOM.A.F.S-162][NOM.A.M.P-13][NOM.D.F.S-153][NOM.G.F.S-154][NOM.N.F.P-14][NOM.N.F.S-162] [NOM.N.M.P-13][NOM.N.M.S-13]
11	Fibr	osen	2.05564283054928	[ADJ.EN-139][NOM.A.F.P-159][NOM.A.M.P-15][NOM.A.M.S-14][NOM.D.F.P-159][NOM.D.M.P-17][NOM.D.M.S-14][NOM.G.F.P-159] [NOM.G.F.S-12][NOM.G.M.P-15][NOM.G.M.S-14][NOM.N.F.P-159][NOM.N.M.P-15]
11	Schles	ien	2.00515936981159	[ADJ.EN-19][NOM.A.F.P-1987][NOM.A.M.P-18][NOM.A.X.P-364][NOM.A.X.S-11][NOM.D.F.P-1989][NOM.D.M.P-29][NOM.D.X.P-366] [NOM.D.X.S-11][NOM.G.F.P-1987][NOM.G.F.S-44][NOM.G.M.P-17][NOM.G.X.P-364][NOM.N.F.P-1987][NOM.N.M.P-18][NOM.N.X.P-364] [NOM.N.X.S-11][VER.IN-24][VER.PI.ILS-23][VER.PI.NOUS-23][VER.SUP.ILS-53][VER.SUP.NOUS-53][VER.ZU-12]
11	Schles	ier	1.76973731216616	[ADJ.ER-17][NOM.A.M.P-82][NOM.A.M.S-133][NOM.A.X.S-36][NOM.D.M.S-133][NOM.D.X.S-36][NOM.G.M.P-82][NOM.N.M.P-82] [NOM.N.M.S-134][NOM.N.X.S-36][VER.IM.TU-842][VER.PI.JE-841]
11	Schles	isch	0.43315332023955	[ADJ.0-1077][NOM.A.M.S-13][NOM.D.M.S-13][NOM.N.M.S-13][VER.IM.TU-36][VER.PI.JE-30]
11	Schles	ische	0.634456566757471	[ADJ.E-1077][NOM.A.M.P-14][NOM.G.M.P-11][NOM.N.M.P-14][VER.IM.TU-31][VER.PI.JE-31][VER.SUP.IL-31][VER.SUP.JE-31]
11	Schles	ischen	0.696312463412551	[ADJ.EN-1077][NOM.D.M.P-14][VER.IN-30][VER.PI.ILS-31][VER.PI.NOUS-31][VER.SUP.ILS-31][VER.SUP.NOUS-31][VER.ZU-17]
11	Schles	ischer	0.689009238476659	[ADJ.COMP.0-900][ADJ.ER-1080]
11	Schles	isches	0.0646779394943541	[ADJ.ES-1077][NOM.G.M.S-13]
11	Schles	ke	2.43233966637074	[ADJ.E-21][NOM.A.F.S-159][NOM.A.M.P-112][NOM.A.X.P-42][NOM.D.F.S-158][NOM.G.F.S-159][NOM.G.M.P-110][NOM.G.X.P-42] [NOM.N.F.S-159][NOM.N.M.P-112][NOM.N.M.S-43][NOM.N.X.P-42][VER.IM.TU-515][VER.PI.JE-512][VER.SUI.IL-26][VER.SUI.JE-26] [VER.SUP.IL-512][VER.SUP.JE-512]
11	Schles	ser	2.44819701868427	[ADJ.COMP.0-181][ADJ.ER-223][NOM.A.M.P-101][NOM.A.M.S-101][NOM.A.X.P-28][NOM.A.X.S-11][NOM.D.M.S-101][NOM.D.X.S-11] [NOM.G.F.P-11][NOM.G.M.P-112][NOM.G.X.P-28][NOM.N.M.P-101][NOM.N.M.S-113][NOM.N.X.P-28][NOM.N.X.S-11][VER.IM.TU-21] [VER.PI.JE-21]
11	Kreuz	chen	2.93155909129782	[ADJ.EN-1599][NOM.A.F.P-171][NOM.A.M.P-44][NOM.A.M.S-43][NOM.A.X.P-116][NOM.A.X.S-116][NOM.D.F.P-171][NOM.D.F.S-12] [NOM.D.M.P-203][NOM.D.M.S-43][NOM.D.X.P-141][NOM.D.X.S-122][NOM.G.F.P-171][NOM.G.F.S-14][NOM.G.M.P-44][NOM.G.M.S-30] [NOM.G.X.P-116][NOM.N.F.P-171][NOM.N.M.P-44][NOM.N.M.S-15][NOM.N.X.P-116][NOM.N.X.S-116][VER.II.ILS-130][VER.II.NOUS-130] [VER.IN-531][VER.PI.ILS-532][VER.PI.NOUS-532][VER.PPS.0-128][VER.SUI.ILS-130][VER.SUI.NOUS-130][VER.SUP.ILS-532] [VER.SUP.NOUS-532][VER.ZU-304]
11	Kreuz	iger	1.44402118467646	[ADJ.COMP.0-563][ADJ.ER-747][NOM.A.M.P-36][NOM.A.M.S-36][NOM.D.M.S-36][NOM.G.F.P-38][NOM.G.M.P-83][NOM.N.M.P-36] [NOM.N.M.S-82]

Annexe H. Schéma général d'acquisition

Les impératifs industriels n'ayant laissé aucun espace d'expérimentation, le schéma suivi pour la construction des ressources fut le même pour toutes les langues. Nous distinguons les quatre grandes étapes suivantes[124].

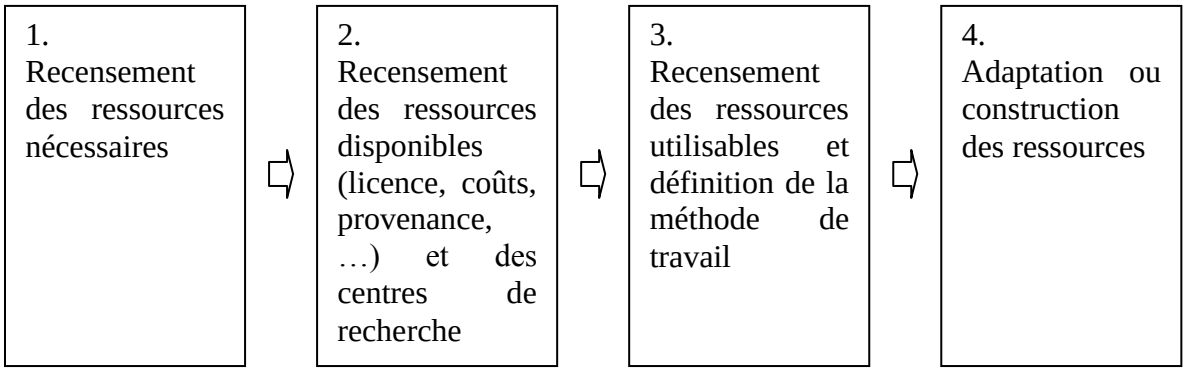


Figure 91 : Grandes étapes dans la phase d'acquisition d'un ensemble de ressources

1. Recensement des ressources nécessaires : Selon les traitements envisagés, les ressources nécessaires sont différentes. Pour les traitements faits dans le moteur de Sinequa, il s'agit principalement de lexiques morphosyntaxiques de large couverture, de corpus étiquetés morphosyntaxiquement et de grammaires de détection d'entités.

2. Recensement des ressources disponibles : Ce recensement revient à dresser un état de l'art des ressources et des traitements semblables à ceux que l'on veut mettre en place. Il ne se restreint pas à un passage dans les catalogues d'ELRA et du LDC, car de nombreuses ressources n'y sont pas présentes. En même temps on identifie les centres de recherche actifs sur le domaine afin de nouer des contacts si nécessaire.

3. Recensement des ressources utilisables et définition de la méthode de travail : On dresse la liste des ressources qui peuvent être exploitées pour la construction en suivant le schéma de la figure 92 ci-dessous. Dépendant de la liste ainsi obtenue, la méthode de construction de chaque ressource est définie dans le respect du budget global.

4. Adaptation et/ou construction des ressources : Après obtention des ressources, les ressources sont reformatées, intégrées, adaptées selon les besoins de l'application.

[124] Dans ce schéma nous avons négligé la phase de recrutement de la personne qui dispose des compétences nécessaires pour effectuer ces travaux. Devant avoir une double compétence en linguistique et en informatique en plus d'être de langue maternelle de la langue à développer, cette personne n'est pas toujours facile à trouver malgré les nombreuses formations existantes.

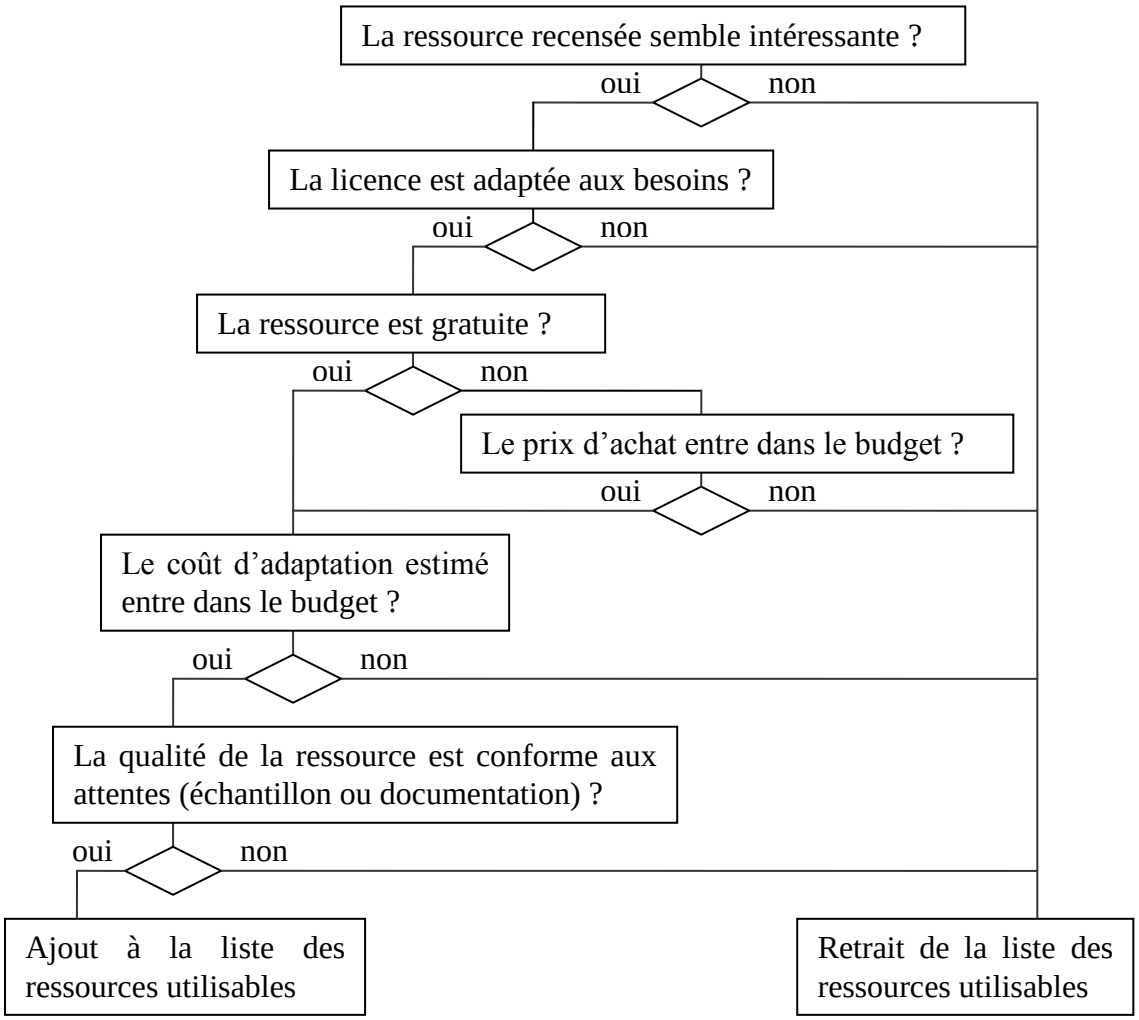


Figure 92 : Construction de la liste des ressources utilisables

L'obtention d'un échantillon est absolument nécessaire pour vérifier le rapport qualité/prix de la ressource, car la description est rarement complète. Pour remédier à ce manque de description, nous faisons une proposition de normalisation dans l'Annexe I.

La liste des ressources utilisables est en général assez courte, et les choix en découlent assez naturellement. La stratégie d'acquisition se décide en effet en fonction de ces ressources. S'il existe bien une pénurie de ressources pour certaines langues, nous n'avons pas été confronté à une absence totale de ressources pour les langues développées sauf pour le finnois. L'absence de ressources exploitables était à l'époque totale. La jonction des particularités de la langue finnoise nous a contraint à changer la gestion habituelle du lexique. L'approche adoptée est détaillée dans la partie 9.1.

La bonne disponibilité de ressources pour les langues que nous avons développées s'explique par le fait qu'il s'agit de langues qui représentent un intérêt commercial à court ou moyen terme pour Sinequa. Nous nous trouvons donc en dehors des travaux faits sur les *langues en danger* dont la caractéristique est notamment d'être peu dotée en ressources. Nous avons aussi remarqué que la disponibilité des ressources n'est pas forcément en relation avec la taille de la communauté : les ressources pour le catalan, le galicien et le basque sont bien plus accessibles que celles de l'italien.

Annexe I. Proposition pour normaliser la description des ressources

Le recensement des ressources disponibles est une phase qui n'est pas toujours évidente du fait du manque d'un dépôt central des descriptions des ressources. Quand les descriptions sont disponibles, elles sont rarement complètes, mêmes si elles émanent de professionnels de la distribution des ressources. Pour chaque type de ressource, nous donnons les clés pour une bonne description. Ces méta-informations qui sont à la fois quantitatives et qualitatives peuvent être utilisées pour dresser une fiche standard pour chaque type.

1. Corpus

Pour la description d'un corpus, on mentionnera les informations suivantes :

- taille des fichiers : la taille moyenne des fichiers, les tailles minimale et maximale.

- format/s des fichiers : le ou les formats des fichiers et le nombre de fichiers dans chaque format s'il y a plusieurs formats.

- nombre de mots : Pour quantifier un corpus, le nombre de mots est plus parlant que la taille des fichiers, mais il est moins évident à obtenir. Cela demande en effet une définition normalisée de ce qu'est un mot. Dans ce contexte on utilise plus souvent le terme *token*, pour lequel nous proposons la définition suivante : un *token* est soit une ponctuation soit une série de caractères non séparés par un espace ni par une ponctuation. Il existe néanmoins des langues ou des écritures qui n'utilisent pas les mêmes mécanismes pour séparer les mots, comme le thaï ou le chinois. Pour ces langues il faut passer par un traitement de plus haut niveau pour obtenir le nombre de mots. Comme pour la taille, on donnera le nombre de mots moyen, minimal et maximal des fichiers. Une description détaillée donnera le nombre de mots pour chaque fichier.

- raison de constitution : Elle indique pourquoi le corpus a été constitué, et donne en cela une indication de l'utilisation potentielle comme ressource. Il en découle aussi les critères de sélection des fichiers, qu'il faut exprimer de façon claire.

- sources : site web, livres, magazines, etc. avec au moins l'année d'édition pour donner une idée du niveau de langage. Si la moitié d'un corpus est constitué d'un livre scientifique en linguistique des années 70, il ne conviendra guère à l'apprentissage d'un modèle générique.

- origine : Il s'agit le plus souvent d'internet, d'un partenaire de projet de recherche, d'un client, d'un centre de distribution de ressources, d'une université, etc. S'il a été obtenu par téléchargement, il faut indiquer avec quel outil l'aspiration s'est faite, ou bien préciser qu'il a été téléchargé manuellement.

- traitements : Si le corpus a subi des traitements, il faut les décrire (par exemple : nettoyage, conversion, enrichissement) et indiquer les méthodes utilisées s'il n'y a pas de référence disponible. S'il y a ajout de métadonnées, il faut décrire leur champ d'application (par exemple : le fichier, le mot, la structure, etc.).

- licence : Elle exprime les conditions d'utilisation, des restrictions fortes pouvant exister si le corpus provient d'un client ou d'un partenaire d'un projet de recherche.

- liens avec d'autres ressources ou traitements : Il est possible qu'il existe des ressources liées d'une manière ou d'une autre au corpus, typiquement un lexique qui aurait servi pour l'annotation du corpus, ou un modèle de langage entraîné à partir du corpus.

2. Lexique

Pour la description d'un lexique, on mentionnera les informations suivantes :

- nombre de lemmes, de mots-formes, de mots composés et de noms propres : Ces chiffres sont de bons indicateurs du contenu du lexique, mais doivent être interprétés avec précaution comme nous le verrons un peu plus loin.

- nombre de lemmes décliné selon la catégorie grammaticale : La comparaison du nombre de lemmes dans une même catégorie grammaticale entre deux lexiques de langues apparentées donne de bonnes indications sur les proportions entre les catégories et le nombre de lemmes.

- méthode de constitution et documentation : La méthode de constitution donne des indications sur la fiabilité des informations, permettant de comprendre d'éventuelles erreurs de codage. En l'absence de système normalisé pour les informations linguistiques, une documentation des étiquettes morphosyntaxiques est par exemple indispensable.

- liens avec d'autres ressources ou traitements : Il est possible qu'il existe des ressources liées d'une manière ou d'une autre au corpus, typiquement un corpus qui aurait été étiqueté avec le lexique.

Les nombres de lemmes et de mots-formes sont indispensables pour quantifier le lexique, mais sont à interpréter avec précaution. Comme nous pouvons voir dans le tableau 31 ci-dessous, le ratio entre mots-formes et lemmes est très différent d'une langue à l'autre[125]. Les chiffres sont calculés sur des lexiques de grande couverture, ne comportant pas de noms propres. Le nombre de lemmes est d'autant plus comparable que le nombre de mots-formes est donc insignifiant.

EN	NL	DA	EL	SV	FR	IT	DE	ES	PT	PL	RU	FI
1,42	2,69	3,17	3,35	3,74	4,18	4,41	7	7,29	8,94	15,05	18,23	120,15

Tableau 31 : Ratio mots-formes/lemme pour chaque langue (lexiques Sinequa)

Pour mieux comprendre ces chiffres, et pour pouvoir répondre à la question de savoir pourquoi par exemple l'allemand a un ratio moindre que l'espagnol et le portugais en dépit de la présence des cas, nous avons calculé les ratios mots-formes/lemme pour les quatre catégories principales : nom, verbe, adjectif et adverbe.

[125] Ces ratios ne doivent pas être pris comme des valeurs universelles. Les lexiques sur lesquels sont calculés ces chiffres n'ont pas été construits pour étudier la langue mais pour des traitements particuliers qui reposent également sur des grammaires de génération ou d'interprétation morphologique en complément.

	Nom	Adjectif	Verbe	Adverbe
DA	3,84	2,37	5,93	1,00
DE	2,62	12,12	22,69	1,01
EL	2,74	7,88	3,28	1,00
EN	1,81	1,11	4,02	1,01
ES	2,14	3,33	48,79	1,00
FI	32,14	95,69	409,93	1,00
FR	2,02	2,62	33,96	1,00
IT	1,89	3,62	40,78	1,00
NL	1,31	4,11	11,18	1,00
PL	8,93	10,45	54,70	1,07
PT	2,40	3,93	54,71	1,02
RU	9,02	15,09	42,02	1,00
SV	3,58	4,48	5,20	1,02

Tableau 32 : Ratios mots-formes/lemme par catégorie grammaticale pour chaque langue

On voit donc bien que les langues à cas comme l'allemand (avec le russe, le polonais et le finnois) introduisent beaucoup plus de mots-formes par lemme pour les noms et les adjectifs. Néanmoins, la conjugaison semble bien plus réduite en allemand qu'en espagnol ou en portugais, la proportion inégale des différentes catégories et le nombre différent de lemmes faisant le reste.

Pour mieux visualiser les ratios mots-formes/lemme, nous avons projeté ces ratios sur des lexiques de 76 495 lemmes, ce qui était la moyenne du nombre de lemmes de l'ensemble des lexiques de large couverture en janvier 2009. Nous avons respecté la répartition moyenne des lemmes sur les quatre premières catégories sur l'ensemble des lexiques pour réaliser cette projection. Elle est représentée dans le tableau 33. En prenant les moyennes nous espérons lisser le fait qu'il peut tout naturellement exister dans une langue une surreprésentation d'une catégorie par rapport aux autres langues.

	Nb lemmes	
Nom	45 385	59 %
Adjectif	16 500	20 %
Verbe	9 649	13 %
Adverbe	3 977	6 %

Tableau 33 : Parts moyennes des catégories dans les lexiques de large couverture de Sinequa toutes langues confondues

Cette projection est artificielle et approximative, mais illustre tout de même la grande différence entre les langues en ce qui concerne le nombre d'informations rien qu'en prenant en compte le nombre de mots-formes. Dans l'histogramme nous faisons abstraction du finnois avec ses 6 997 266 mots-formes qui rendrait illisible notre histogramme en écrasant les autres valeurs.

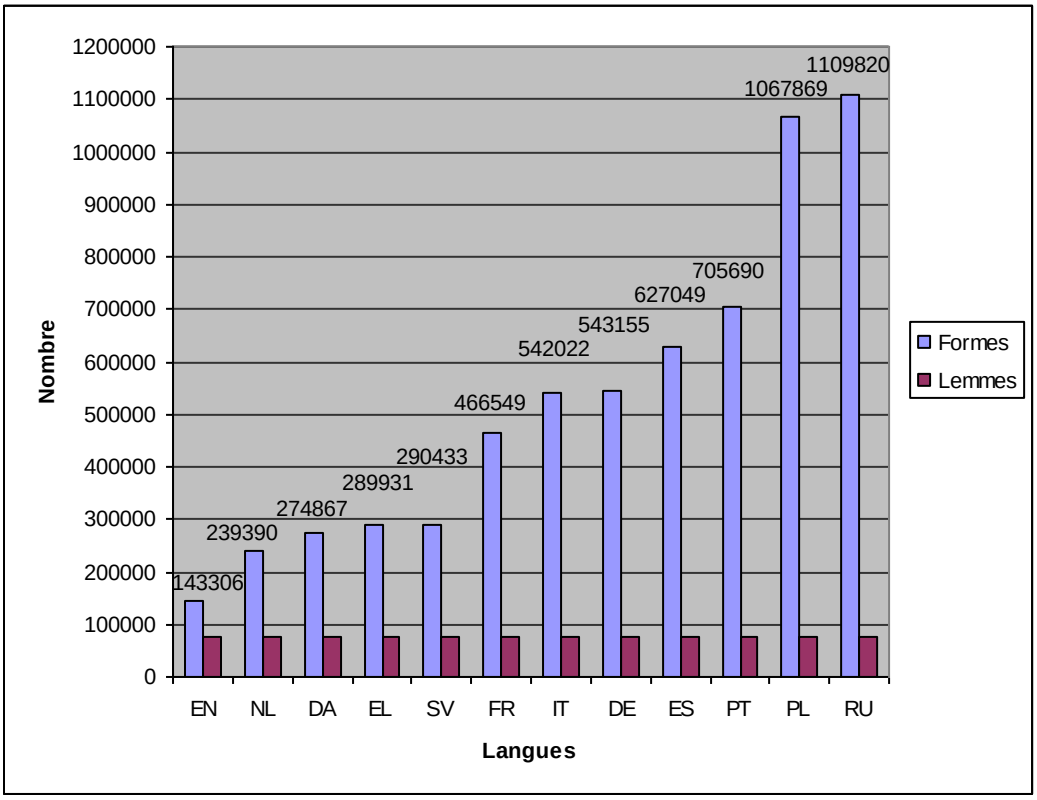


Tableau 34 : Nombre de mots-formes par langue à nombre de lemmes égal (76 500)

La connaissance du nombre de noms propres faisant partie du lexique est également importante, et cela pour deux raisons : les noms propres peuvent assez facilement gonfler le nombre de lemmes d'un lexique, et comme ils ne se fléchissent pas, influencer le ratio mots-formes/lemme. Il existe d'autres mots avec les mêmes propriétés, comme les abréviations et les sigles. Dans l'idéal, il faudrait indiquer également leurs nombres.

Une remarque importante est qu'il faut prendre comme base de calcul pour le lemme les combinaisons <lemme-catégorie grammaticale> unique. Selon le format du dictionnaire, il est possible que les lemmes ne désignent pas des entrées uniques. Ainsi, *avoir* en français peut être nom ou verbe. Les deux entrées auront le même lemme, mais seront comptabilisées deux fois en prenant les combinaisons <lemme – catégorie grammaticale>.

3. Grammaires

Pour la description d'une grammaire, on mentionnera les informations suivantes :

- méthode de constitution et documentation : La méthode de constitution donne des indications sur la fiabilité des informations, permettant de comprendre d'éventuelles erreurs.

- liens avec d'autres ressources ou traitements : Il est possible qu'il existe des ressources liées d'une manière ou d'une autre au corpus, typiquement un corpus qui aurait été étiqueté avec le lexique.

- origine : L'origine donne des informations sur les raisons de constitution de la grammaire et d'éventuels déploiements dans des systèmes.

Annexe J. Ressources génériques ou spécialisées ?

Dans l'état de l'art, les meilleurs résultats des traitements de l'écrit sont obtenus quand les ressources sont totalement en phase avec les documents analysés. Les corpus à l'origine des lexiques sont homogènes, or il n'est pas économiquement viable de développer des ressources différentes pour chaque type de document. En revanche, il est possible, en acceptant une certaine dégradation des résultats, de mettre en place des traitements robustes dont les résultats sont suffisants sur plusieurs types de documents. Dans ce chapitre, nous examinons les propriétés des différents types de documents et en tirons des conclusions prudentes sur la possibilité de partager des ressources pour traitement des différents types.

1. Une volonté industrielle de ressources générales

La contrainte industrielle qui nous a été imposée est d'avoir une seule configuration de ressources pour le traitement de tout type de texte indexé par le moteur de recherche, quel que soit le type du texte ou son format. Les ressources doivent donc être générales dans le sens où elles permettent de traiter tout texte.

Au niveau du lexique, cela se traduit par une agrégation des lexiques de langue non marquée et des langues spécialisés. Cette langue générale est aussi l'agrégat des particularités des langages régionaux. Les variations entre le français de la France, de la Belgique et du Canada se trouvent ainsi mélangés, ainsi que celles du néerlandais des Pays-Bas et de la Belgique néerlandophone[126] ou de l'anglais américain et de l'anglais britannique. Les différences peuvent être significatives au niveau du vocabulaire et des constructions syntaxiques, même s'il s'agit souvent de préférences et que les différentes constructions syntaxiques sont valides sur la totalité du territoire linguistique. Les différences syntaxiques pourraient complexifier les automates de détection d'entités nommées, mais en général l'écrit se conforme plus qu'on ne le pense au standard en vigueur.

Une gestion rationnelle des ressources impose des ressources à même d'être déployées dans toutes les circonstances. Les traitements sont robustes, avec un taux d'erreurs acceptable, qui est de zéro pour certains traitements.

Ces dernières cinq années, ce modèle a vacillé au moins deux fois. La première fois faisant suite à une adaptation des grammaires de détection d'entités à du corpus tout venant, alors qu'elles avaient originalement été construites et testées sur du corpus journalistique. Les erreurs en sortie des automates originaux étaient majoritairement dues à de mauvaises conversions de document Excel et PDF, suite à quoi les automates ont été refaits. Le prix à payer pour une meilleure robustesse était une chute du nombre de détections. Cela n'a rien d'étonnant : l'accroissement de la précision est accompagné d'une diminution du rappel. Cette baisse de performance sur du corpus journalistique est difficile à assumer par rapport aux clients dans la presse. La décision de mettre en place des grammaires dédiées amènerait à une déclinaison du produit selon les différents secteurs. Cette décision ne nous appartient évidemment pas.

L'autre moment critique a eu lieu lorsque nous avons traité des transcriptions de langage conversationnel dans le cadre du projet Infom@gic. La nécessité d'appliquer un modèle de langage particulier paraissait évidente et les expériences en ont montré l'intérêt. Nous avons

[126] Pour le néerlandais, quelques exemples significatifs sont décrits dans Schuurman et al. (2003) à l'occasion de la constitution du corpus du néerlandais parlé CGN qui couvre le néerlandais parlé au Pays-Bas et en Belgique.

donc un nouveau modèle d'étiquetage morphosyntaxique dédié au langage conversationnel. Quand il sera intégré au produit dans un avenir proche, il faudra bien mettre en place plusieurs distributions intégrant des ressources différentes selon le produit vendu.

Dans d'autres situations, notamment l'indexation des flashs infos radiophoniques (projet Audiosurf), aucune adaptation ne s'est révélée nécessaire, et l'impact sur les fonctionnalités avancées était même très limité comme nous avons constaté dans [Cailliau et Loupy, 2007].

Nous avons travaillé sur une multitude de types de documents et de langues. Certains types présentent des caractéristiques qui sont plus proches d'un texte bien écrit, d'autres sont plus proche du langage parlé. Comme les traitements sont plus adaptés à un type de corpus qu'à un autre quel que soit le modèle en vigueur, il est important de savoir en quoi les documents traités diffèrent exactement. Cela nous amènera à décider quelles sont les conditions qui demandent un traitement spécifique, et donc la mise en place d'une distribution spécifique.

Dans ce qui suit, nous allons essayons de déterminer dans quelles situations une distribution spécifique s'impose.

2. Oral transcrit – texte écrit : une fausse dichotomie

Les systèmes de communication de l'écrit et de l'oral sont très différents. [Shriberg, 2005] décrit les principaux défis à relever pour arriver à un traitement automatique efficace de l'oral. Il s'agit de l'absence de ponctuation, le phénomène de la disfluence, les tours de parole superposés et la détection de l'émotion. Cette dernière fait partie de ces dimensions pragmatiques qui se perdent quand on en fait des transcriptions et qui sont indispensables pour une bonne compréhension de la parole. Il en va de même pour les éléments extra-verbaux (gestes, expressions) des locuteurs, et l'environnement, qui participent activement à la conversation, ne serait-ce que par la deixis.

Tous les éléments pragmatiques ne sont pas également présents dans les différents genres oraux. Ceux où les éléments pragmatiques sont moins explicites, comme par exemple dans les émissions radio, se rapprochent de l'écrit : nombre réduit de disfluences (surtout dans les bulletins d'information), explicitation par éléments verbaux de l'environnement si nécessaire, une linéarité de la parole en général bien respectée (donc peu de parole superposée[127]).

Si on prend la transcription de conversation en situation et le texte bien écrit (par exemple l'article de journal) comme les deux genres prototypiques de respectivement l'oral et l'écrit, alors on peut classifier les différents types de documents selon les caractéristiques de l'un et de l'autre. Si on ajoute aux caractéristiques de Shriberg la présence d'éléments déictiques sans référent textuel et qu'on classe six genres de documents on obtient le tableau 35. Les informations concernant les textes bien écrits, les flashs infos, interviews radiophoniques, et conversations téléphoniques sont issues de nos observations sur corpus. Pour les propriétés du langage SMS, nous nous sommes appuyé sur les observations et expériences de [Anis, 2002], [Fairon et al., 2006], [Guimier de Neef et al., 2007] et [Kobus et al., 2008].

Notre classification se fait à partir de phénomènes qui posent problème pour le traitement automatique des langues dans le but de savoir quand on peut raisonnablement utiliser les mêmes ressources. Nous ne cherchons donc pas à caractériser finement des genres ou des

[127] Les enregistrements de conversations téléphoniques se font aujourd'hui toujours sur un seul canal, faute de matériel industriel sur le marché capable d'enregistrer sur deux canaux différents.

types de texte à partir d'un grand nombre de marqueurs linguistiques dans la lignée des travaux de D. Biber[128] qui travaille sur ce sujet depuis les années 80.

Types de documents	Ponctua-tion	Présence de disfluences	Emotion, opinion explicite textuelle	Super-position des tours de parole	Déixis sans référent textuel
Texte bien écrit (article de journal, recette de cuisine, soluce de jeu, blog, e-mail)	*Oui*	*Non*	*Oui*	*Non*	*Non*
Transcription de flash infos radiophonique (présence de tours de parole)	**Non**	*Non*	*Oui*	*Non*	*Non*
Conversions de pdf, word, excel,…	**Non**	*Non*	*Oui*	*Non*	*Non*
Texte mal écrit (sms, blog, e-mail, …)	**Non**	*Non*	*Oui*	*Non*	**Oui**
Conversation écrite (chat informel)	**Non**	**Oui**	*Oui*	*Non*	**Oui**
Transcription de conversation téléphonique, d'interview radiophonique	**Non**	**Oui**	**Non**	**Oui**	*Non*
Transcription de parole conversationnelle	**Non**	**Oui**	**Non**	**Oui**	**Oui**

Tableau 35 : Types de documents et leurs caractéristiques

On voit que les les types de documents partagent plus ou moins de caractéristiques de l'oral (en rouge gras) et de l'écrit (en vert italique). Cela change graduellement lorsqu'on passe en revue les différents types de documents. Ainsi, on peut représenter les types de documents sur un axe horizontal selon le nombre de caractéristiques qu'ils partagent avec l'une des extrémités, qui sont les textes bien écrits et la transcription de parole conversationnelle, tel que nous l'avons établi dans le tableau 35. Nous avons choisi le mot *pragmatique* car c'est le concept le plus proche de ce que nous cherchons à représenter.

[128] Voir [Biber, 2004] et ses références, mis en perspective dans le rapport [LIPN-Paris13, 2007], réalisé dans le cadre du projet Textcoop.

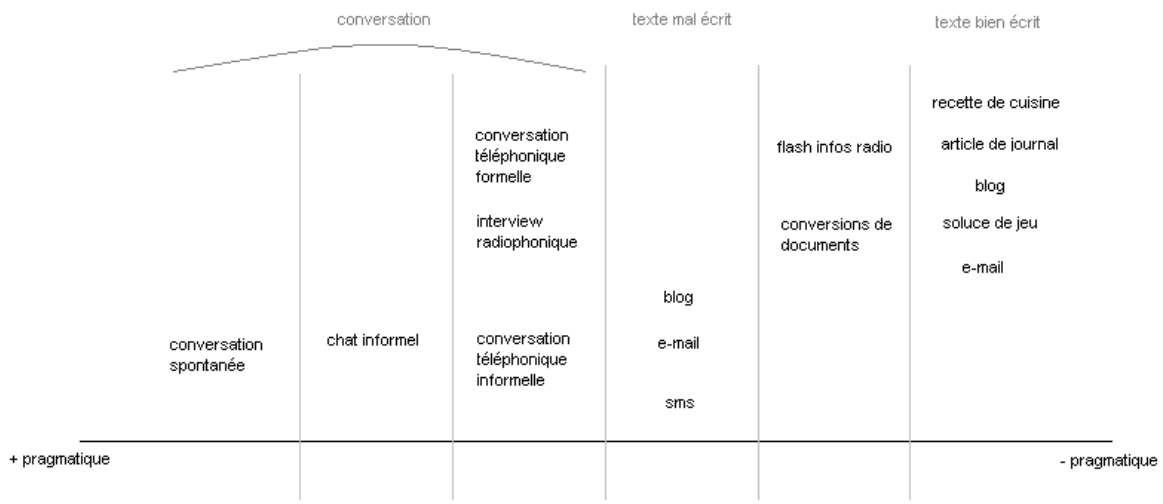


Figure 93 : Types de documents sur l'axe pragmatique

L'axe est représenté en figure 93. Ainsi, les transcriptions d'émissions radio que nous avons étudiées dans le projet Audiosurf, portent toutes les caractéristiques de l'écrit, sauf pour la ponctuation. L'échantillon étudié était quasiment entièrement constitué de bulletins d'information, qui se préparent d'ailleurs comme un texte écrit comme on peut le lire sur une fiche éducative concernant les préparatifs des émissions sur France Inter ([Paulin, 1999]).

Plus proche de l'oral se situent les conversations téléphoniques, qui sont de la parole spontanée. Elles ont toutes les caractéristiques d'une conversation normale, sauf que les éléments sans référent hors conversation sont extrêmement limités. A cause de l'absence de signes non soniques dans la communication, on y recourt à la langue pour expliciter tout ce qui n'est pas visible. Cela les différencie d'une conversation spontanée en situation qui s'appuie fortement sur la pragmatique : elle utilise des signes de nature diverse (mots, gestes, expressivité) et beaucoup d'éléments déictiques sans référent textuel.

Aujourd'hui, le TAL ne s'occupe pas des éléments extratextuels : cela dépasse son objet d'étude actuel qui est la langue écrite, transcrite ou parlée. La compréhension totale de l'ensemble des signes échangés est pourtant une question primordiale à résoudre pour construire le sens complet des énoncés. Imaginons par exemple la conversation en situation suivante : « Regarde (+ geste qui pointe un avion) !) – Oh là là ! » ou bien un père qui dit à son enfant qui tient un objet dangereux dans la main : « Donne-moi ça ». Il est impossible d'interpréter les énoncés oraux sans disposer de toutes les informations. Dans le premier exemple, on ignore qu'il s'agit d'un avion et ce qu'il a ou fait de spécial, et on ne sait pas interpréter la réaction de l'interlocuteur comme admirative ou bien comme catastrophée.

3. Le degré de codification

Une autre dimension à prendre en compte pour le traitement est le degré de codification du type de document. La codification est le degré selon laquelle une production ressemble à la production type en ce qui concerne le style, la structuration et les thèmes abordés. Plus la codification est élevée, plus la production est prototypique. Nous proposons de le mesurer par les critères suivants :

- **Richesse du vocabulaire** : Plus la richesse du vocabulaire est réduite, plus la codification est grande. Les conclusions des études suivantes nous confortent dans

notre analyse. [Gijsel et al., 2005] ont étudié le corpus CGN[129] d'un point de vue sociolinguistique en utilisant le TTR (*Type Token Ratio*) comme mesure de richesse lexicale et montrent une corrélation entre registre, niveau d'éducation et sexe d'une part. D'après leurs recherches, le registre détermine le TTR : l'oral informel, c'est-à-dire dialogique ou spontané, a des TTR moins élevés que l'oral formel, c'est-à-dire monologique ou préparé. Un TTR plus élevé indique un vocabulaire plus riche. Sur des corpus de transcriptions anglais et arabe [Creutz et al., 2007] note une grande différence dans le nombre de mots uniques entre des transcriptions de texte lu et de conversation spontanée. D'après les courbes de l'anglais le vocabulaire est deux fois plus grand pour le texte lu et la différence augmente légèrement avec une plus grande taille de corpus. En arabe l'écart est grand mais reste stable entre les deux types de données.

- **Diversité syntaxique** : Moins la diversité syntaxique est grande, plus la codification est élevée. Nous n'avons pas trouvé de références qui mesurent la diversité syntaxique. Si la complexité d'un modèle de langage d'un tagger est mesurable, on pourrait peut-être comparer les modèles selon les entrainements faits sur des corpus de différents genres, tout en tenant compte des marges d'erreurs.
- **Structuration répétitive** : Plus la structuration entre textes du même genre est répétitive, plus le degré de codification est élevé. Pour l'instant, nous ne connaissons pas de logiciel qui détecte les similarités structurelles entre les documents.

En l'absence de chiffres concrets et de travaux sur le sujet, nous proposons la classification suivante (figure 94) qui est faite à partir du bon sens en intégrant les critères susmentionnés.

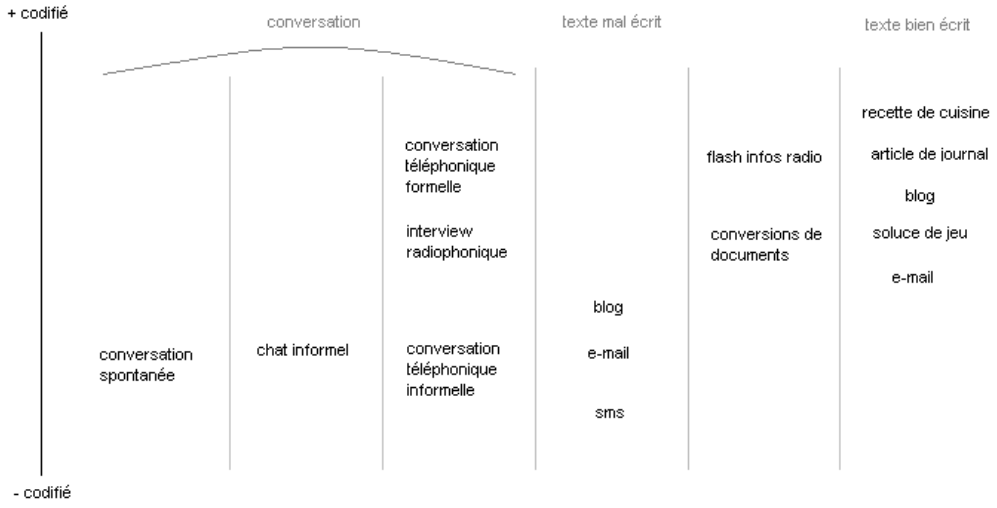


Figure 94 : Types de documents sur l'axe de codification

4. Des ressources en commun pour des types proches

Si nous combinons les deux axes, la complexité pragmatique en X et le degré de codification en Y, nous pouvons situer les différents types de document dans cet espace à deux dimensions. Nous joignons la notion de facilité de traitement aux extrémités des axes, et celle-ci est directement liée aux traitements et méthodes actuellement en vigueur ainsi qu'à la présence explicite ou non des éléments à traiter.

[129] CGN : « Corpus Gesproken Nederlands », corpus du néerlandais parlé. Voir [Schuurman et al., 2003] et [Schuurman et al., 2004] pour plus d'informations.

De gauche à droite, on trouve dans les types de documents un nombre décroissant d'éléments situationnels, de signes non verbaux ou déictiques sans référent textuel à prendre en compte pour une compréhension parfaite. La difficulté de traitement liée à la présence de ces éléments décroît également de gauche à droite.

De haut en bas le degré de codification baisse, et avec lui l'homogénéité d'une production à l'autre du même genre. Les éléments pris en compte sont le style (diversité des structures syntaxiques, richesse du vocabulaire) et la structuration (découpage régulier en sous-parties). Les documents et leur contenu deviennent moins homogènes, ce qui se traduit par une augmentation de la difficulté de traitement.

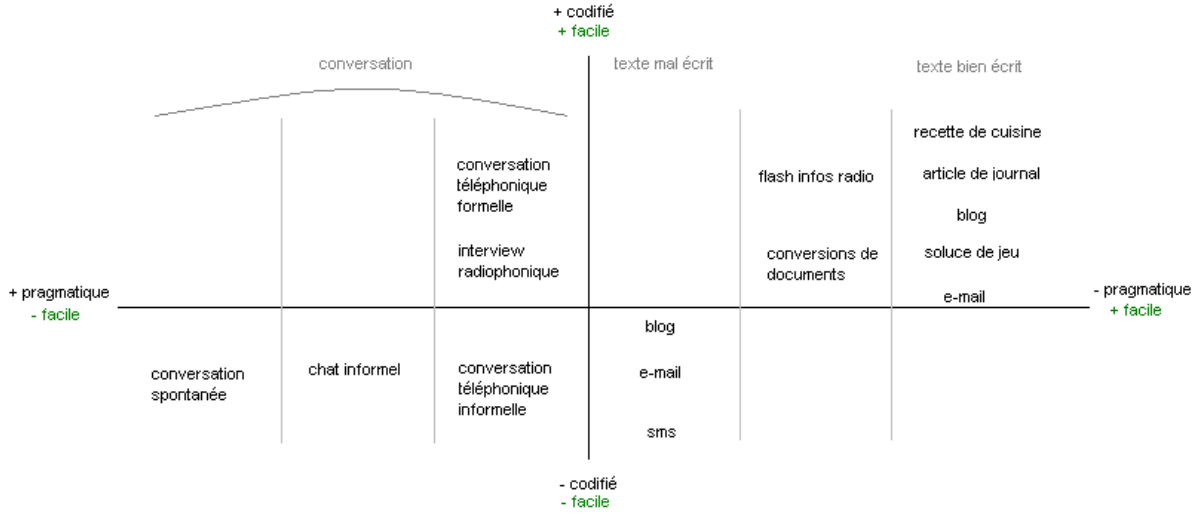


Figure 95 : Types de documents sur le plan codification et pragmatique

Si nous divisons l'espace en quatre quadrants, nous obtenons des ensembles de types de documents avec des propriétés linguistiques et documentaires proches. Pour les types d'un même quadrant, les résultats des traitements sont sans aucun doute suffisants pour partager toutes ou une bonne partie des ressources. Cela correspond effectivement à notre expérience, car nous avons dû développer des ressources spécifiques pour les conversations téléphoniques entre agent et client, alors que les transcriptions de flashs info pouvaient être traitées sans aucun problème avec les ressources des journaux (avec une légère dégradation liée aux erreurs de transcription [Cailliau et Loupy, 2007]).

Il faudra donc prévoir la gestion active de plusieurs distributions, c'est-à-dire de plusieurs ensembles de ressources pour les mêmes traitements en fonction du type de données. La gestion s'en trouve fortement complexifiée, car si certaines ressources partagées continuent à évoluer sans les autres, il faut faire attention à ne pas créer des incompatibilités entre traitements de types différents.

Le quadrant le plus proche de l'origine nous semble actuellement impossible à traiter : l'absence de l'analyse automatique de la dimension pragmatique de l'oral rend impossible le traitement efficace de ce type de productions.

Dans quelque temps, les corpus mélangeront sans doute plusieurs types de documents, obligeant l'analyseur à pouvoir changer de distribution dynamiquement et détecter le type s'il n'est pas donné.

Annexe K. Liste des abréviations de langue

AR	arabe
DA	danois
DE	allemand
EL	grec
EN	anglais
ES	espagnol
FI	finnois
FR	français
IT	italien
JA	japonais
KO	coréen
NL	néerlandais
PL	polonais
PT	portugais
RU	russe
SV	suédois
TH	thaï
ZH	chinois

Des ressources aux traitements linguistiques : le rôle d'une architecture linguistique

Mise en place d'un environnement de gestion de ressources linguistiques pour une plate-forme d'analyse textuelle

Les systèmes intégrant des traitements venant du traitement automatique des langues reposent souvent sur des lexiques et des grammaires, parfois indirectement sur des corpus. A cause de la quantité et de la complexité des informations qu'elles contiennent, ces ressources linguistiques deviennent facilement une source d'incohérence. Dans cette thèse, nous explorons les moyens d'améliorer la gestion des nombreuses ressources linguistiques d'un moteur de recherche industriel en dix-neuf langues qui fait appel à une analyse textuelle élaborée. Nous proposons une méthode pour formaliser l'architecture linguistique des traitements linguistiques et des ressources utilisées par ceux-ci. Cette formalisation explicite la façon dont les connaissances contenues dans les ressources sont exploitées. Grâce à elle, nous pouvons construire des outils de gestion qui respectent l'architecture du système. L'environnement ainsi mis en place se concentre sur la mise à jour et l'acquisition des ressources linguistiques, leur exploitation étant figée par des contraintes industrielles.

Mots clés : architecture linguistique, ressource linguistique, gestion de ressources linguistiques, système de TAL, traitement automatique des langues

The Role of a Linguistic Architecture in Language Processing and its Resources

Establishing an Environment to Manage Linguistic Resources for a Text Analysis Platform

Systems integrating natural language processing often use lexicons and grammars, sometimes indirectly corpora. Because of the quantity and the complexity of the information in these linguistic resources, they are likely to become a source of inconsistency. In this thesis we explore how to improve the management of linguistic resources for an industrial search engine in nineteen languages that performs an elaborate textual analysis. We propose a method to formalize the linguistic architecture of the linguistic processing and its resources. This formalization shows how the knowledge contained in the resources is exploited and gives us the possibility to build management tools compliant with the system's architecture. The environment implemented in this way focuses on updating and acquiring the linguistic resources, while their exploitation is defined by the industrial constraints.

Keywords: linguistic architecture, linguistic resource, linguistic resource management, NLP system, NLP tool, natural language processing

Discipline : Informatique

Laboratoire d'Informatique de Paris-Nord – LIPN – UMR CNRS 7030

Institut Galilée – Université Paris 13 – Paris-Nord

99, avenue Jean-Baptiste Clément, 93430 Villetaneuse